# KINETIC PARAMETER ESTIMATION IN PREPACKAGED FOODS SUBJECTED TO DYNAMIC THERMAL TREATMENTS

By

BRUCE ARI WELT

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

1996

# ACKNOWLEDGMENTS

The author would like to express sincere gratitude to his major advisor, Dr. Arthur A. Teixeira and unofficial co-chairman Dr. Murat O. Balaban for their confidence and enthusiasm throughout this research program. Their friendship, guidance and support were critical ingredients to the successful completion of this work.

The author also wishes to thank his supervisory committee Dr. Glen H. Smerage, Dr. David E. Hintenlang and Dr. Burrell J. Smittle for their ample guidance and suggestions related to the research and resulting technical papers which have been submitted for publication. Likewise, the author wishes to thank graduate coordinator, Dr. K.V. Chau for his friendship and guidance throughout the graduate program and for his assistance with the work covered in Chapter 3.

A special thanks goes to the faculty and staff of the Agricultural and Biological Engineering Department and Department of Food Science and Human Nutrition for assistance with everything from operation of photocopy machines to pilot plant equipment. Special thanks also go to Dr. Daniel S. Sage for his assistance in designing the formal mathematical proof to support the work covered in Chapter 2.

The author wishes to thank his parents, Martin and Ruth Welt, his brother and sister, Andrew and Jodi Welt, and his grandmothers Syd Welt and Pauline Schact, for their love and support through all of the best and worst of times.

Finally, the author wishes to thank his fiancée, Janet E. Evans, for her support and devotion throughout what will be remembered as the beginning of our journey through life together.

## TABLE OF CONTENTS

Abstract of Dissertation Presented to the Graduate School of the University of Florida in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

# KINETIC PARAMETER ESTIMATION IN PREPACKAGED FOODS SUBJECTED TO DYNAMIC THERMAL TREATMENTS

By

Bruce Ari Welt

August, 1996

Chairman: A. A. Teixeira
Major Department: Agricultural and Biological Engineering

Consumer demand for higher quality processed foods provides an impetus for designing processes that are less detrimental to product quality, while maintaining adequate levels of security from food borne illness. Thermal process design requires knowledge kinetic parameters of food constituents that affect product safety and quality.

The purpose of this study was to develop a method for estimating product and process specific thermal kinetic parameters. The method, referred to as the Paired Equivalent Isothermal Exposures (PEIE) method, may be applied to products that heat non-uniformly under non-isothermal heating conditions. Such conditions are typical with thermally processed prepackaged foods.

The research was performed in four phases: (1) development of the PEIE method for the case of foods and/or samples that heat uniformly, (2) assessment of approaches to heat transfer simulation for foods that do not heat uniformly, (3) development of the PEIE method for the case of foods that do not heat uniformly and (4) demonstration of one potential application of the PEIE method involving hybrid irradiation/thermal process design.

The PEIE method was used to quantify the impact of irradiation pretreatment on the thermal sensitivity of *Bacillus stearothermophilus* (NCA 1518) spores in canned pea puree. Cans of inoculated puree were irradiated with x-rays at 0, 1.0 and 3.0 kGy at an electron linear accelerator facility prior to thermal treatment in a vertical still-cook retort. Enhanced thermal sensitivity was apparent from changes in Arrhenius kinetic parameters that described the rate of inactivation of the spores as a function of temperature. Mean rate constants, $k_{121.1}$, at 121.1°C and activation energy values, $E_a$, for spores in pea puree pretreated with 0, 1.0 and 3.0 kGy were 0.26, 0.28 and 0.38 $min^{-1}$ and 250, 200 and 200 kJ/mol, respectively. Thermal processes designed to provide a 5 log cycle reduction in spore count were shown to require $F_o$ values of 44.5, 41.0 and 30.0 minutes, respectively. Retort process simulations indicated that irradiation offered the potential for significant thermal process reductions. Steam shut-off times of 72, 65 and 54 minutes, corresponding to products treated with 0.0, 1.0 and 3.0 kGy, respectively, were estimated for subsequent thermal treatment at 121.1°C.

# CHAPTER 1
# INTRODUCTION

Thermal processing is one of the most widely used methods of food preservation today. The principal objective of thermal processing is microbial inactivation by heat. Thus, thermal process design requires consideration of microbial inactivation kinetics and heat transfer principles. Since foods are very sensitive to environmental factors, especially temperature, relatively poor product quality is often the consequence of thermal processing. Consumer demand for higher quality processed foods provides an impetus for designing processes that are less detrimental to product quality, while maintaining adequate levels of security from food borne illness.

Improving thermal processing equipment, operational techniques, and/or process design methods can contribute to higher quality thermally processed foods. Significant improvements in equipment and operational techniques have been made throughout this century. Relatively recent developments in computer automation technology have also contributed. However, methods of designing thermal processes remain virtually unchanged.

Although it is now generally known that microbial inactivation kinetic parameters are functions of various external factors, the thermal processing industry continues to treat such parameters as immutable (Perkins et al., 1975). Examples of external factors that influence inactivation kinetics include (1) conditions during sporulation, (2) conditions of spore recovery, (3) physical treatment of spores, and (4) the particular chemistry of the suspending medium. Since microbial inactivation parameters are product and process specific, efforts should be made to understand how such conditions affect thermal processing requirements. Additionally, methods such as irradiation are available that enhance sensitivity of microorganisms to heat. Accurately accounting for sensitized organisms in thermal process design could allow for significant reductions in thermal process severity. Therefore, methods are required that allow determination of product and process specific thermal kinetic parameters.

The purpose of this study was to develop a method for estimating thermal kinetic parameters on a product and process specific basis. This approach requires a departure from traditional procedures, because it relies on survivor experiments performed in-situ. Since foods may or may not heat uniformly and processes are seldom isothermal, such methods need to account for such situations.

The research was performed in four phases: (1) development of the method for the case of uniformly heating foods, (2) assessment of approaches for heat transfer simulation in foods that do not heat uniformly, (3) development of the method for the case of foods that do not heat uniformly, and (4) application of the method for designing thermal treatments when a nonuniformly heating product was pretreated with relatively mild doses of ionizing radiation. Each phase of the work is reported separately in chapters 2, 3, 4 and 5. Overall concepts are progressively built from chapter to chapter, however, each chapter was prepared as a complete conceptual unit.

The purpose of this format was to facilitate submission of the chapters, as independent entities, to scientific journals for hopeful publication as refereed articles. The benefit of this format is that each chapter may be read independently. This comes at the cost of some redundancy when covering the work in its entirety, since each chapter contains independent background material. Chapter 6 provides conclusions and implications relevant to the entire work and all references were compiled into one section at the end.

# CHAPTER 2
# KINETIC PARAMETER ESTIMATION IN FOODS THAT HEAT UNIFORMLY

## Introduction

The Equivalent Point Method (EPM) was developed by Swartzel (1982) in order to facilitate design of continuous flow ultra-high temperature (UHT) processes. Subsequently, it was suggested that the EPM offered a reliable means for determining kinetic parameters from dynamic thermal processing conditions (Swartzel, 1984). The EPM has been used in kinetic analysis and thermal assessment of continuous flow systems (Wescott et al., 1995; Berry, et al., 1990; Sadeghi and Swartzel, 1990; Swartzel, 1986; Swartzel and Jones, 1985). The EPM was recognized as, "a significant advance in the application of food technology to food production" (Giese, 1994, p. 94), which contributed to earning the Industrial Achievement Award from the Institute of Food Technologists in 1994. This recognition is a testament to the value of practical non-isothermal methods for kinetic parameter estimation to the food industry.

Two approaches are commonly used to obtain thermal kinetic parameters that involve either isothermal or non-isothermal experiments (Lenz and Lund, 1980). The isothermal approach is used more often, due to conceptual simplicity. However, the approach is often criticized for unavoidable thermal lags, inconveniently small sample sizes and a limited temperature range from which to select sufficiently different isothermal conditions (Welt et al., 1993). In fact these problems confound each other since thermal lags become more prevalent as either sample size or selected isothermal temperature increases.

The isothermal approach may be preferable when the goal is reaction order determination (Constantinou, 1994). Although reactions of primary interest in food processing often involve complex mechanisms and multiple pathways, it is well known that most of these reactions follow *apparent* first-order kinetics with Arrhenius temperature dependence (Labuza, 1979). Since thermal process design often involves the assumption of first-order kinetics with Arrhenius temperature dependence, there is room for expanded use of the non-isothermal approach in kinetic parameter estimation.

Non-isothermal approaches for kinetic parameter estimation are attractive because they are specifically designed to deal with thermal transients. This not only eliminates much of the experimental tedium associated with the isothermal approach, but also allows for determination of kinetic parameters from realistic processing conditions. Since food chemistry is very sensitive to thermal exposure and food chemistry influences the kinetic behavior of food constituents (mircoorganisms), kinetic parameters obtained from realistic processing conditions may be more appropriate for use in food process design. The use of non-isothermal methods is limited, however, due to increased complexity of data analysis. Several non-isothermal methods for kinetic parameter estimation have been described (Nasri et al., 1993; Lenz and Lund, 1980; Hayakawa et al., 1969).

As proposed by Swartzel (1984) the EPM represented an easy-to-use non-isothermal method for kinetic parameter estimation. Consistent with the name, the method suggests that an equivalent isothermal process may be obtained for any dynamic thermal process. This concept, in-and-of-itself, is not new to food engineers, as processes designed to provide a specified sterilizing value (number of log cycles that the concentration of the target organism is reduced) are designed to achieve a specific $F_o$ value (equivalent time at 121.1°C for a given target microorganism) (Teixeira, 1992).

The $F_o$ value provides a means to equate a dynamic process to an equivalent isothermal process for a particular target organism. The isothermal process may be described as an idealized rectangular pulse, where the product is instantaneously and uniformly heated to the isothermal temperature, held at this temperature for a certain amount of time, and then instantaneously and uniformly cooled to a non-lethal (non-reacting) temperature. If we say the $F_o$ of a dynamic process is 5 minutes, we are saying that the lethality (extent of reaction) of the target organism (reactant) is equivalent to what would have been observed under isothermal conditions at 121.1°C (250°F) for 5 minutes. Notice that two pieces of information are required in order to specify an $F_o$ value, (1) a particular reactant represented by its kinetic parameters and (2) an arbitrarily specified time or temperature. Selection of the temperature 121.1°C (250°F) is a matter of adopted convention, since this is the temperature at which many industrial sterilizers are operated.

Mathematically, the only difference between the EPM and $F_o$ value method is that the EPM does not involve an arbitrarily specified time or temperature. Thus, the method must provide a means to satisfy this additional degree of freedom. The EPM appears to be much more powerful, however, since it is claimed to be able to specify an isothermal process that is universally applicable to all reactants that follow Arrhenius kinetics. This universality was implied by the assumption that the equivalent point “is independent of the $E_a$ [activation energy] value” (Swartzel, 1984, p. 804). Careful review of the development of the EPM in this work, however, revealed that this assumption is not valid, suggesting that the EPM should not be used for kinetic parameter estimation without further consideration.

The goal of this work was, (1) to describe and mathematically demonstrate the basis for questioning the assumption related to the universality of an equivalent point, (2) to develop an alternative approach for kinetic parameter estimation from non-isothermal conditions, and (3) to demonstrate the new method with sample calculations.

## Theory

The functional form of a first-order rate process is

$$\frac{dC}{dt} = -kC \qquad (2\text{-}1)$$

where C is the concentration of a reactant at time, t, and k is the reaction rate constant. Solving Eq. 2-1 by integration gives

$$\frac{C}{C_0} = e^{-k(t-t_0)} \qquad (2\text{-}2a)$$

or equivalently,

$$\ln(\frac{C}{C_0}) = -k \cdot (t - t_0) \tag{2-2b}$$

The temperature dependence of the rate constant, k, may be described by the Arrhenius equation,

$$k = k_0 \cdot \exp\left\{\frac{-E_a}{R \cdot T}\right\} \tag{2-3}$$

where kinetic parameters $k_o$ and $E_a$ are the pre-exponential factor and activation energy, respectively, R is the ideal gas law constant and T is absolute temperature. Under isothermal conditions, the extent of a reaction may be determined directly by substituting Eq. 2-3 into Eq. 2-2:

$$\ln(\frac{C}{C_0}) = -k_0 \cdot \exp\left\{\frac{-E_a}{R \cdot T}\right\} \cdot (t - t_0). \tag{2-4}$$

In order to determine the extent of a reaction under non-isothermal conditions Eq. 2-4 becomes

$$\ln(\frac{C}{C_0}) = -k_0 \cdot \int_{t_0}^{t} \exp\left\{\frac{-E_a}{R \cdot T(t)}\right\} dt. \tag{2-5}$$

At this point the EPM deviates from the $F_o$ value method. In order to determine the $F_o$ value for a dynamic process with respect to a target organism, the right-hand side of Eq. 2-5 is equated to the right-hand side of Eq. 2-4. The isothermal temperature, T, in Eq. 2-4 is arbitrarily set to 394.25 Kelvin (121.1°C), the known activation energy, $E_a$, of the target organism is substituted on both sides, and the resulting equation is solved for time, t, from Eq. 2-4. The equivalent time at 121.1°C is referred to as the $F_o$ value of the dynamic process, with respect to the particular target organism ($E_a$ value).

In a similar manner, the EPM equates Eq. 2-5 to Eq. 2-4, but introduces the extra step of normalizing the resulting equation by $-k_o$. The resulting equation takes the form

$$\frac{\ln(\frac{C}{C_0})}{-k_0} = G = \int_{t_o}^{t} \exp\left\{\frac{-E_a}{R \cdot T(t)}\right\} dt = t_e \cdot \exp\left\{\frac{-E_a}{R \cdot T_e}\right\} \tag{2-6}$$

where G, as defined by Swartzel (1982), is the value obtained by evaluating the integral from the actual thermal history. As with the $F_o$ value method, the time, $t_e$, at isothermal temperature, $T_e$, comprises the equivalent isothermal exposure which would provide the same extent of reaction (lethality), as the dynamic thermal process.

Swartzel (1982) observed that straight-line equations of the form

$$\ln(t_e) = \ln(G) + \frac{E_a}{R \cdot T_e} \tag{2-7}$$

resulted when these G values were equated to the isothermal part of Eq. 2-6 (right-hand side). Using several of these equations from arbitrarily selected $E_a$ values, Swartzel (1982) plotted the lines on an $\ln(t_e)$ versus $1/T_e$ plot and noticed that the lines had a tendency to intersect in a particular region of the plot. The effect was sufficiently convincing that Swartzel (1982) claimed that all such lines intersect at one point. Swartzel (1984, p. 804) noted situations where three or more lines did not intersect at the same point, but attributed this to experimental uncertainty and loss of arithmetic precision during calculations:

> Any accumulated error in the process (thermocouple calibration, data collection, determination of exponential integral, etc.) would be indicated with less than perfect intersections with three or more log(t) versus T lines. The intersection points of these lines may be evaluated by regression analysis to determine the most error free statistical intersection point for all lines.

Although the EPM specifies the need for more than one activation energy to define the equivalent process, algebra dictates that not more than two are actually necessary. Since an isothermal process is functionally represented as

$$G = t_e \cdot \exp\left\{\frac{-E_a}{R \cdot T_e}\right\}, \tag{2-8}$$

two sets of $E_a$ - G values provide two equations in two unknowns, $T_e$ and $t_e$, which may be solved as follows:

$$\begin{array}{rl} \ln(G_1) = & \ln(t_e) - \dfrac{E_{a1}}{R \cdot T_e} \\ -\ln(G_2) = & -\ln(t_e) + \dfrac{E_{a2}}{R \cdot T_e} \\ \hline \end{array} \tag{2-9a}$$

$$T_e = \frac{(E_{a2} - E_{a1})}{R \cdot \ln({}^{G_1}{}_{G_2})}. \tag{2-9b}$$

Equivalent time, $t_e$, is found by back-substitution of the equivalent temperature, $T_e$, into either equation of Eq 9a.

In order for the equivalent point to be independent of $E_a$, all additional equations would have to provide the same solution. To see why this is not possible, consider Eq. 2-6, Eq. 2-7 and Eq. 2-9. Equation 6 shows that for a given actual thermal history, T(t), G is a monotonic function of $E_a$; each different value of $E_a$ yields a different and distinct value of G. With $E_a$ and G defined, substitution into Eq. 2-7 yields one unique equation with two unknowns, $t_e$ and $T_e$. Equation 9 shows that $t_e$ and $T_e$ may be specified from two different $E_a$ - G versions of Eq. 2-7 by solving them simultaneously. Since the two equations in Eq. 2-9a are independent and unique, their solution, or point of intersection, must also be unique. It is not possible for Eq. 2-9 to yield the same solution for any other pairs of independent equations derived from different $E_a$ values. A rigorous proof of this statement was provided in Appendix A. Thus, it is not possible to specify a unique equivalent isothermal process from a dynamic thermal exposure that is universally applicable to all reactants.

An isothermal process derived from two $E_a$ values may be interpreted as the equivalent exposure that would result in the same extents of reaction (lethalities) for reactants (microorganisms) characterized by the two respective $E_a$ values. An isothermal process specified in this manner may be referred to as an equivalent isothermal exposure (EIE). The definition of an EIE differs from the $F_o$ value. The $F_o$ value provides an equivalent time at a given temperature, such that the same extent of reaction (lethality or sterilizing value) would be observed for reactants (microorganisms) characterized by the one $E_a$ value used to determine the $F_o$ value. An EIE specifies an isothermal exposure (time and temperature) that would provide equivalent extents of reaction (lethality) for reactants characterized by the two $E_a$ used to define the EIE.

## EPM Example

The following example illustrates the dependence of the equivalent point on activation energy. As proposed, the EPM involves the following steps:

1. Record the dynamic temperature history.
2. Arbitrarily select several $E_a$ values and determine corresponding G values by evaluating the integral in Eq. 2-6, using the actual thermal history.
3. Equate G values to the isothermal form of the equation (right most side of Eq. 2-6), using respective $E_a$ values.
4. Generate straight line curves on a $\ln(t_e)$ versus $1/T_e$ plot, for each $E_a$ - G equation.
5. Obtain the equivalent isothermal process from a unique intersection point.

Step 5 is the point of contention, as it shall be demonstrated that the lines generated in step 4 do not intersect at one point.

For the purpose of this example, we assumed that a food item was uniformly and non-isothermally heated from 0°C to 140°C over a total process time of 200 minutes, according to the following linear equation (see Fig. 2-1, Process a):

$$T(\text{Kelvin}) = 273 + 0.7t \qquad (2\text{-}10)$$

where T is in Kelvin and t is in minutes. Activation energy, $E_a$, values of 30, 100 and 400 kJ/mol were arbitrarily selected for equivalent point calculations. The following integral was solved in order to calculate G values for each $E_a$ value:

$$G = \int_{0}^{200} \exp\left\{\frac{-E_a}{R \cdot (273 + 0.7t)}\right\} dt \qquad (2\text{-}11)$$

The integrals were numerically evaluated to a tolerance of 1 x $10^{-6}$, using the programmable spreadsheet program, Excel (Microsoft Corp., Redmond, WA). An integration algorithm of Simpson's rule (Press et al., 1992) was used to program the spreadsheet using Visual Basic (Microsoft Corp., Redmond, WA). Calculated G values are shown in Table 2-1.

By plotting the three respective forms of Eq. 2-7, three points of intersection, representing three different isothermal processes, were obtained for each pair of $E_a$ values (i.e. for $E_a$ value pairs 1-2, 2-3 and 1-3). The equivalent isothermal processes obtained were 392.4 Kelvin for 87.8 minutes, 404.0 Kelvin for 67.5 minutes, and 406.8 Kelvin for 29.7 minutes, relating to activation energy pairs of 30 and 100, 30 and 400, and 100 and 400 kJ/mol, respectively. The spatial relationship of these intersection points is illustrated in Fig. 2-2. This example shows that one unique intersection point should not be expected to exist.

Thermocouple errors were not a factor here, as this was a mathematical exercise. Additionally, the same points of intersection were obtained when integration was performed to a tolerance of 0.01, indicating that precision of numerical integration and rounding errors did not alter the result significantly. Therefore, this example supports the conclusion that a unique equivalent isothermal process, independent of activation energy, should not be expected to exist. This suggests that kinetic parameter estimates obtained with the EPM may not be accurate. Use of inaccurate kinetic parameter estimates could result in insufficient thermal process specifications.

## Development of the Method of Paired Equivalent Isothermal Exposures (PEIE)

A method, referred to as Paired Equivalent Isothermal Exposures (PEIE), was developed for estimating kinetic parameters from dynamic thermal processes using the true nature of an equivalent isothermal exposure (EIE). For a given dynamic thermal exposure, the point of intersection of any two $E_a$ - G lines on a $\ln(t_e)$ versus $1/T_e$ plot, specifies an EIE (time, $t_e$, at temperature, $T_e$) with respect to reactants characterized by the two particular $E_a$ values used to specify the EIE. The method of PEIE exploits this definition to allow estimation of thermal kinetic parameters of a reactant that has been subjected to at least two dynamic thermal exposures. Since two $E_a$ values are required to define an EIE from one dynamic thermal exposure, when one of the two $E_a$ values is the apparent $E_a$ value of the reaction, the corresponding isothermal rate constant, $k^T$, when plotted on an Arrhenius plot, must fall somewhere along the desired, but unknown Arrhenius curve. The other $E_a$ value simply defines the location of this point along the Arrhenius curve. Thus, if the reaction-apparent $E_a$ value is used to define EIE's from any pair of dynamic thermal exposures, the desired Arrhenius curve may be specified. The crux of the method of PEIE, therefore, is to find the reaction-apparent $E_a$ value for all pairs of data related to dynamic thermal exposures. The method of PEIE's is iterative in nature, whereby an initial $E_a$ guess is increasingly improved such that corresponding EIE's result in points that converge to the appropriate Arrhenius curve.

Two examples of the PEIE method are provided to demonstrate (1) that the PEIE method is capable of finding the exact parameters used to generate simulated experiments and (2) that the method is capable of finding the best possible parameters using uncertain, but realistic extent-of-reaction data. The following steps outline the method of PEIE:

1. Record temperature histories and initial and final concentrations of a reactant from at least two dynamic thermal processes that result in different extents of reaction.

2. Arbitrarily select one $E_a$ value, $E_a1$, and establish a scheme for specifying the other $E_a$ value, $E_a2$. For each iteration, the value of $E_a2$ may be a fixed value or some multiple of $E_a1$. Theoretically, the choice of $E_a2$ does not matter, as is the case when extent-of-reaction data are exactly known. Implications of various $E_a2$ specification schemes for the typical case of uncertain extent-of-reaction data are discussed below.

3. Determine respective EIE's for each pair of dynamic thermal experiments using the $E_a$ values specified in step 2.

4. Calculate isothermal rate constants, $k^T$, with Eq. 2-2b using the EIE specifications (from step 3) and respective extent-of-reaction data (from step 1).

5. For each pair of $k^T$ values determined in step 4, simultaneously solve Eq. 2-3, to find $E_a$ values of curves that pass through respective pairs of points. Graphically, each $k^T$ represents a point on an Arrhenius plot. Values of $E_a$ are determined from the slopes of curves that pass through each pair of points (e.g. if three dynamic thermal experiments were performed, calculate $E_a$ estimates for each of the pairs 1-2, 1-3, and 2-3).

6. Return to step 3 and replace the old $E_a1$ values with respective $E_a$ estimates obtained in step 5. Repeat the procedure until $E_a1$ stops changing.

Create an Arrhenius plot in the typical fashion with $k^T$ values from all final EIE specifications. Obtain kinetic parameters in the typical fashion by performing a regression analysis of all $k^T$ points on the Arrhenius plot.

## PEIE Example #1

The following example was designed to demonstrate the ability of the PEIE method to determine kinetic parameters that were used to generate extent-of-reaction data from dynamic thermal experiments. Consider experiments involving thiamin in pea puree using the three dynamic thermal exposures depicted in Fig. 2-1. Expected extents of thiamin degradation from each exposure may be calculated by substituting representative kinetic parameters into Eq. 2-5 ($E_a$ = 113 kJ/mol and $k_o$ = 1.0E+13 $\text{min}^{-1}$, Welt et al., 1993). Extents of thiamin degradation were calculated and shown in the column labeled 'Exact (C/Co)' in Table 2-2. The PEIE method was applied to these data as follows:

Step 1: See Table 2-2.

Step 2: For the first iteration, $E_a1$ was chosen to be 30 kJ/mol. A scheme which specified $E_a2$ as 2 x $E_a1$ was selected (see Table 2-3).

Step 3: Equivalent isothermal exposures for each process pair (a-b, a-c and b-c) were determined as previously described (see columns labeled "$T_e$" and "$t_e$" in Table 2-3).

Step 4: Isothermal rate constants, $k^T$, were calculated using EIE and final concentration data. These calculations are illustrated below for the first iteration, and summarized in Table 2-3:

$$\begin{aligned} k_a^{386.435} &= -\ln(0.421)/101.194 = 8.548 \text{ x } 10^{-3} \text{ min}^{-1} \\ k_b^{377.662} &= -\ln(0.600)/135.230 = 3.773 \text{ x } 10^{-3} \text{ min}^{-1} \\ k_c^{368.882} &= -\ln(0.773)/160.974 = 1.600 \text{ x } 10^{-3} \text{ min}^{-1} \end{aligned} \tag{2-12}$$

Step 5. Activation energy, $E_a$, estimates were calculated for each pair of $k^T$ values by substitution into respective Arrhenius equations (Eq. 2-3). Solving these simultaneously yields Eq. 2-13

$$E_a = \frac{R \cdot \ln\left({}^{k_1}/_{k_2}\right)}{\left(\dfrac{T_1 - T_2}{T_1 \cdot T_2}\right)} \tag{2-13}$$

where the subscripts, 1 and 2, refer to different dynamic thermal processes.

Step 6: The $E_a$ estimates were used to replace respective $E_a1$ values. $E_a2$ values were appropriately adjusted and the second iteration followed by repeating the procedure from step 3 (Table 2-4). Details from the third and final iteration are shown in Table 2-5.

Step 7: Regression analysis was performed on $\ln(k^T)$ versus $1/T_e$ data from the final iteration (Table 2-5). Resulting kinetic parameters were $E_a = 113,000$ J/mol and $k_o = 1.00 \times 10^{13}$ min$^{-1}$.

Table 2-6 shows the progress of each iteration (see 'Iteration #1 - 3,' Table 2-6). Although all calculations were performed with full precision (15 digit) arithmetic, not all digits are shown in the tables. Simpson's rule was used to calculate the G value integral in Eq. 2-6 with a tolerance of $1 \times 10^{-6}$ (Press et al., 1992). Improvements to the parameters were indicated by reductions to the sum of squared error (SSQ error) of predicted data (Table 2-6).

PEIE Example #2

The same dynamic thermal processes (Fig. 2-1) and kinetic parameters ($E_a$ = 113 kJ/mol, $k_o$ = 1.0E+13 min$^{-1}$) were used as a basis for this example. However, experimental uncertainty was introduced in the calculated final concentration data. For each data point, random values were selected from an evenly distributed range specified by the exact final concentration (column labeled 'Exact,' Table 2-2), plus or minus 5 percent. Equations of the thermal histories and *observed* extent-of-reaction data were provided in Table 2-2 (column labeled 'Observed,' Table 2-2).

For the first iteration, a value of 300 kJ/mol was arbitrarily chosen for $E_a1$. As with the previous example, $E_a2$ was set to 2 x $E_a1$ for each iteration. Details of all iterations are provided in Tables 7, 8 and 9. Progress of iterations was provided in Table 2-10. The best parameters given the *observed* data were $E_a$ = 120,289 J/mol and $k_o$ = 9.270E+13 $min^{-1}$.

## Discussion

The fact that the method of Paired Equivalent Isothermal Exposures (PEIE) was capable of precisely determining kinetic parameters used to generate the data in example #1 provides confidence that the method works as proposed. An absolute performance comparison to the Equivalent Point Method (EPM) (Swartzel; 1982, 1984) is not possible, because contrary to the fundamental assumption of the EPM, the result depends on the $E_a$ values used to determine an *equivalent point*. Although this work presented reasons to question the fundamental assumption regarding the universality of equivalent points, the mechanics of applying the EPM to kinetic parameter estimation, while not identical, are similar to those of the first iteration of the PEIE method. With the EPM, if a researcher has a preconception of the true $E_a$ value and/or is fortunate enough to select a value that is close to the true value, then the EPM may provide parameters that are close to what might be considered correct parameters. This provides an explanation for the apparent success of the EPM in the literature. A notion of the differences in results obtained from the EPM and PEIE method can be seen by comparing results of the first and last iterations of the PEIE method. For example #1, Arrhenius curves that provided kinetic parameters from the first and last PEIE iterations are shown in Fig. 2-3. The magnitude of these differences may be assessed by comparing extent-of-reaction predictions from these sets of parameters to the *exact* experimental data in Table 2-6. The SSQ errors obtained using parameters from the first and last iterations were 6.79E-02 and 1.93E-14, respectively, indicating that the PEIE method provided closer agreement with exact experimental data. Figure 2-4 and Table 2-10 show that the same was true for example #2.

The purpose of example #2 was to demonstrate, (1) how experimental uncertainty introduces additional points on the equivalent Arrhenius plot (Fig. 2-4), and (2) how these points introduce an additional consideration regarding selection of a scheme for specifying $E_a2$ for each iteration. For the first iteration, the same values of $E_a1$ and $E_a2$ were used for all pairs of experiments (Table 2-7). The resulting EIE for process *a* when paired with process *b* was the same as the EIE for *a* when paired with *c*. Thus, only three points appeared on the Arrhenius plot from the first iteration (Fig. 2-4). Since there was uncertainty in the extent-of-reaction data, $E_a1$ and $E_a2$ values differed for each pair of experiments in subsequent iterations. Therefore, the EIE for process *a* when paired with process *b*, was different from the EIE for *a* when paired with *c*. Since three dynamic thermal processes were considered in the example, six points appeared on the final Arrhenius plot ('Iteration #3,' Fig. 2-4).

Ideally, the value of $E_a2$ should not influence the result. Figure 2-5 is an Arrhenius plot for example #1, where extent-of-reaction data were known exactly. The figure shows points from final iterations when $E_a2$ was specified according to the following schemes: (1) 10 kJ/mol, (2) $E_a1$ / 2, (3) $E_a1$ x 2 (4) $E_a1$ x 4 and (5) 1,000 kJ/mol. As expected, Fig. 2-5 shows that the same Arrhenius curve was defined regardless of the scheme used to specify $E_a2$. However, in the face of experimental uncertainty, additional points appear on the Arrhenius plot and their spatial relationships may artificially bias the final regression analysis. Figure 2-6 is an Arrhenius plot for example #2 (uncertainty in extent-of-reaction data), showing points from final iterations when $E_a2$ was specified according to the schemes just described. When $E_a2$ was relatively small, EIE's involved low temperatures and long times, resulting in relatively low isothermal rate constants (lower right, Fig. 2-6). Additionally, small $E_a2$ values resulted in points that tended to have greater horizontal scatter (EIE's with larger temperature differences) than those from high $E_a2$ values. Conversely, high $E_a2$ values resulted in points that tended to be more vertically oriented (upper left, Fig. 2-6). Both extremes may artificially influence regression calculations, resulting in poorer quality parameter estimates. As expected, all points in Fig. 2-6 tended to represent one particular Arrhenius curve, however, results from separate regressions show some variation. Table 2-11 provides separate regression results and resulting SSQ error values for all cases considered in Fig. 2-6. The table shows that for the present example the best parameters, as indicated by the smallest SSQ error, were obtained when $E_a2$ was specified as $E_a1$ x 4. Inspection of Fig. 2-6 shows that this scheme resulted in points that were fortuitously oriented with respect to the underlying kinetic behavior.

Although the $E_a1$ x 4 scheme provided the lowest SSQ error value of the $E_a2$ schemes considered here, the $E_a1$ x 2 scheme was chosen for demonstration in order to emphasize the fact that in most cases one reasonable $E_a2$ scheme is sufficient for the

PEIE method to provide adequate kinetic parameter estimates. Comparison of SSQ values indicates marginal improvement in predictive ability of the kinetic parameters. Therefore, the $E_a$1 x 2 scheme was considered to be a useful and easily recallable guideline for kinetic parameter estimation of food constituents with the PEIE method.

The method of PEIE requires data from at least two dynamic thermal processes to generate kinetic parameter estimates. However, when only two experiments are used, no indication of uncertainty in the parameter estimates is evident. As with the traditional isothermal kinetics approach, a minimum of three data points (three non-isothermal experiments) is recommended, as linear regression of the Arrhenius plot helps to mitigate the effect of random uncertainty. Additionally, statistical analysis of the regression provides insight into the reliability of the kinetic parameter estimates. As with any method of kinetic parameter estimation, the ultimate success of the PEIE method in generating reliable kinetic parameters depends on the quality of the original experimental data.

Table 2-1. $E_a$ and corresponding G values calculated from Eq. 2-8.

| Index | Ea Value (J/mol) | G Value |
|---|---|---|
| 1 | 30,000 | 8.914E-03 |
| 2 | 100,000 | 4.277E-12 |
| 3 | 400,000 | 1.281E-50 |

Table 2-2. Hypothetical dynamic thermal experiments used to demonstrate calculations of the Paired Equivalent Isothermal Exposures (PEIE) method.

| Process | Temperature Function (t in min) | Process Time (min) | Exact (C/Co) | Observed (simulated) (C/Co) |
|---|---|---|---|---|
| a | T(K) = 273 + 0.7t | 200 | 0.421 | 0.412 |
| b | T(K) = 273 + 0.5t | 260 | 0.600 | 0.588 |
| c | T(K) = 273 + 0.4t | 300 | 0.773 | 0.793 |

Table 2-3. First iteration of the PEIE method for example #1. Progress from each iteration is shown in Table 2-6.

| Process | Ea | G | Te | te | k | Ea |
|---|---|---|---|---|---|---|
| Pair | Guesses | | | | | Estimate |
| | (J/mol) | | (K) | (min) | (1/min) | (J/mol) |
| a | 30,000 | 8.914E-03 | | | | |
| a | 60,000 | 7.853E-07 | 386.435 | 101.194 | 8.548E-03 | |
| b | 30,000 | 9.590E-03 | | | | |
| b | 60,000 | 5.137E-07 | 377.662 | 135.230 | 3.773E-03 | 113,101 |
| a | 30,000 | 8.914E-03 | | | | |
| a | 60,000 | 7.853E-07 | 386.435 | 101.194 | 8.548E-03 | |
| c | 30,000 | 9.094E-03 | | | | |
| c | 60,000 | 5.137E-07 | 368.882 | 160.974 | 1.600E-03 | 113,162 |
| b | 30,000 | 9.590E-03 | | | | |
| b | 60,000 | 6.801E-07 | 377.662 | 135.230 | 3.773E-03 | |
| c | 30,000 | 9.094E-03 | | | | |
| c | 60,000 | 5.137E-07 | 368.882 | 160.974 | 1.600E-03 | 113,220 |

Table 2-4. Second iteration of PEIE for example #1. Progress from each iteration is shown in Table 2-6.

| Process | Ea | G | Te | te | k | Ea |
|---|---|---|---|---|---|---|
| Pair | Guesses | | | | | Estimate |
| | (J/mol) | | (K) | (min) | (1/min) | (J/mol) |
| a | 113,101 | 8.393E-14 | | | | |
| a | 226,201 | 2.141E-28 | 404.824 | 32.897 | 2.629E-02 | |
| b | 113,101 | 4.947E-14 | | | | |
| b | 226,201 | 1.145E-29 | 395.204 | 43.938 | 1.161E-02 | 113,000 |
| a | 113,162 | 8.240E-14 | | | | |
| a | 226,324 | 2.065E-28 | 404.828 | 32.880 | 2.631E-02 | |
| c | 113,162 | 2.447E-14 | | | | |
| c | 226,324 | 1.145E-29 | 385.579 | 52.306 | 4.923E-03 | 113,000 |
| b | 113,220 | 4.769E-14 | | | | |
| b | 226,441 | 5.181E-29 | 395.212 | 43.895 | 1.162E-02 | |
| c | 113,220 | 2.403E-14 | | | | |
| c | 226,441 | 1.104E-29 | 385.583 | 52.281 | 4.925E-03 | 113,000 |

Table 2-5. Third iteration of PEIE for example #1. Progress from each iteration is shown in Table 2-6.

| Process | Ea | G | Te | te | k | Ea |
|---|---|---|---|---|---|---|
| Pair | Guesses | | | | | Estimate |
| | (J/mol) | | (K) | (min) | (1/min) | (J/mol) |
| a | 113,000 | 8.650E-14 | | | | |
| a | 226,000 | 2.272E-28 | 404.817 | 32.924 | 2.627E-02 | |
| b | 113,000 | 5.103E-14 | | | | |
| b | 226,000 | 1.266E-29 | 395.197 | 43.974 | 1.160E-02 | 113,000 |
| a | 113,000 | 8.650E-14 | | | | |
| a | 226,000 | 2.273E-28 | 404.817 | 32.924 | 2.627E-02 | |
| c | 113,000 | 2.575E-14 | | | | |
| c | 226,000 | 1.266E-29 | 385.569 | 52.375 | 4.916E-03 | 113,000 |
| b | 113,000 | 5.103E-14 | | | | |
| b | 226,000 | 5.921E-29 | 395.197 | 43.974 | 1.160E-02 | |
| c | 113,000 | 2.575E-14 | | | | |
| c | 226,000 | 1.266E-29 | 385.569 | 52.375 | 4.916E-03 | 113,000 |

Table 2-6. Progress of PEIE iterations for example #1.

| | | Predicted from iteration # | | |
|---|---|---|---|---|
| | | 1 | 2 | 3 |
| ko (1/min) | | 1.690E+13 | 9.973E+12 | 1.000E+13 |
| Ea (J/mol) | | 113,162 | 112,990 | 113,000 |
| Process | Exact | | | |
| | (C/Co) | (C/Co) | (C/Co) | (C/Co) |
| a | 0.421 | 0.24846 | 0.42094 | 0.42107 |
| b | 0.600 | 0.44023 | 0.60022 | 0.60035 |
| c | 0.773 | 0.66131 | 0.77289 | 0.77299 |
| SSQ Error | | 6.790E-02 | 4.327E-08 | 1.932E-14 |

Table 2-7. First iteration of PEIE for example #2 ($E_a2 = 2 \times E_a1$). Progress from each iteration is shown in Table 2-10.

| Process | Ea | G | Te | te | k | Ea |
|---|---|---|---|---|---|---|
| Pair | Guesses | | | | | Estimate |
| | (J/mol) | | (K) | (min) | (1/min) | (J/mol) |
| a | 300,000 | 7.541E-38 | | | | |
| a | 600,000 | 4.354E-76 | 409.801 | 13.062 | 6.797E-02 | |
| b | 300,000 | 1.151E-38 | | | | |
| b | 600,000 | 9.489E-80 | 399.952 | 17.426 | 3.044E-02 | 111,153 |
| a | 300,000 | 7.541E-38 | | | | |
| a | 600,000 | 4.354E-76 | 409.801 | 13.062 | 6.797E-02 | |
| c | 300,000 | 1.403E-39 | | | | |
| c | 600,000 | 9.489E-80 | 390.100 | 20.731 | 1.121E-02 | 121,607 |
| b | 300,000 | 1.151E-38 | | | | |
| b | 600,000 | 7.601E-78 | 399.952 | 17.426 | 3.044E-02 | |
| c | 300,000 | 1.403E-39 | | | | |
| c | 600,000 | 9.489E-80 | 390.100 | 20.731 | 1.121E-02 | 131,555 |

Table 2-8. Second iteration of PEIE for example #2 ($E_a2$ = 2 x $E_a1$). Progress from each iteration is shown in Table 2-10.

| Process<br>Pair | Ea<br>Guesses<br>(J/mol) | G | Te<br><br>(K) | te<br><br>(min) | k<br><br>(1/min) | Ea<br>Estimate<br>(J/mol) |
|---|---|---|---|---|---|---|
| a | 111,153 | 1.504E-13 | | | | |
| a | 222,306 | 6.771E-28 | 404.689 | 33.426 | 2.656E-02 | |
| b | 111,153 | 8.994E-14 | | | | |
| b | 222,306 | 6.073E-32 | 395.075 | 44.646 | 1.188E-02 | 111,242 |
| a | 121,607 | 6.580E-15 | | | | |
| a | 243,214 | 1.407E-30 | 405.365 | 30.768 | 2.886E-02 | |
| c | 121,607 | 1.724E-15 | | | | |
| c | 243,214 | 6.073E-32 | 386.068 | 48.934 | 4.748E-03 | 121,681 |
| b | 131,555 | 1.738E-16 | | | | |
| b | 263,110 | 7.910E-34 | 396.244 | 38.195 | 1.389E-02 | |
| c | 131,555 | 7.617E-17 | | | | |
| c | 263,110 | 1.276E-34 | 386.567 | 45.481 | 5.109E-03 | 131,617 |

Table 2-9. Third iteration of PEIE for example #2 ($E_a2 = 2 \times E_a1$). Progress from each iteration is shown in Table 2-10.

| Process | Ea | G | Te | te | k | Ea |
|---|---|---|---|---|---|---|
| Pair | Guesses | | | | | Estimate |
| | (J/mol) | | (K) | (min) | (1/min) | (J/mol) |
| a | 111,242 | 1.465E-13 | | | | |
| a | 222,484 | 6.423E-28 | 404.695 | 33.402 | 2.658E-02 | |
| b | 111,242 | 8.751E-14 | | | | |
| b | 222,484 | 5.800E-32 | 395.081 | 44.613 | 1.189E-02 | 111,242 |
| a | 121,681 | 6.436E-15 | | | | |
| a | 243,362 | 1.347E-30 | 405.369 | 30.751 | 2.887E-02 | |
| c | 121,681 | 1.684E-15 | | | | |
| c | 243,362 | 5.800E-32 | 386.072 | 48.906 | 4.751E-03 | 121,681 |
| b | 131,617 | 1.705E-16 | | | | |
| b | 263,235 | 7.617E-34 | 396.247 | 38.178 | 1.389E-02 | |
| c | 131,617 | 7.470E-17 | | | | |
| c | 263,235 | 1.227E-34 | 386.570 | 45.461 | 5.111E-03 | 131,617 |

Table 2-10. Progress of PEIE iterations for example #2.

| | | Predicted from iteration # | | |
|---|---|---|---|---|
| | | 1 | 2 | 3 |
| ko (1/min) | | 2.262E+14 | 9.267E+13 | 9.270E+13 |
| Ea (J/mol) | | 121,691 | 120,289 | 120,289 |
| Process | Observed<br>(C/Co) | (C/Co) | (C/Co) | (C/Co) |
| a | 0.412 | 0.23422 | 0.40476 | 0.40469 |
| b | 0.588 | 0.44752 | 0.60282 | 0.60277 |
| c | 0.793 | 0.68400 | 0.78533 | 0.78530 |
| SSQ Error | | 6.308E-02 | 3.094E-04 | 3.092E-04 |

Table 2-11. Results of example #2 with various $E_a2$ specification schemes.

| | | Results per Ea2 scheme | | | |
|---|---|---|---|---|---|
| Ea2 = | | 10,000 | Ea1/2 | Ea1 x 4 | 1,000,000 |
| ko (1/min) | | 9.189E+14 | 2.452E+14 | 6.406E+13 | 4.696E+13 |
| Ea (J/mol) | | 127,578 | 123,447 | 119,063 | 118,025 |
| Process | Observed | | | | |
| | (C/Co) | (C/Co) | (C/Co) | (C/Co) | (C/Co) |
| a | 0.412 | 0.36235 | 0.39421 | 0.40565 | 0.40567 |
| b | 0.588 | 0.58340 | 0.60101 | 0.60085 | 0.59856 |
| c | 0.793 | 0.78395 | 0.78874 | 0.78232 | 0.77936 |
| SSQ Error | | 2.520E-03 | 4.763E-04 | 2.982E-04 | 3.161E-04 |

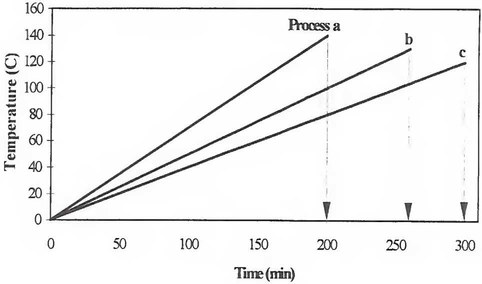


Figure 2-1. Dynamic thermal processes used to demonstrate calculations for the Equivalent Point Method (EPM) and the method of Paired Equivalent Isothermal Exposures (PEIE).

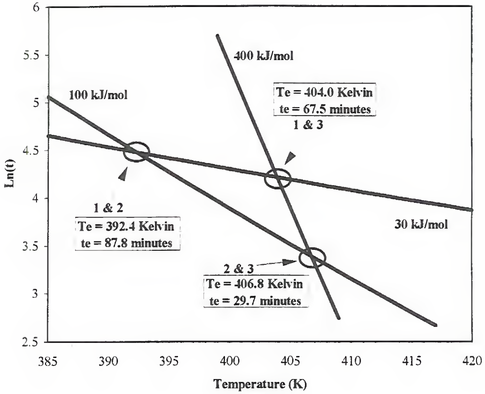


Figure 2-2. Spatial relationship of equivalent isothermal processes, as calculated by the Equivalent Point Method. The figure demonstrates that one unique intersection point, independent of activation energy, should not be expected to exist.

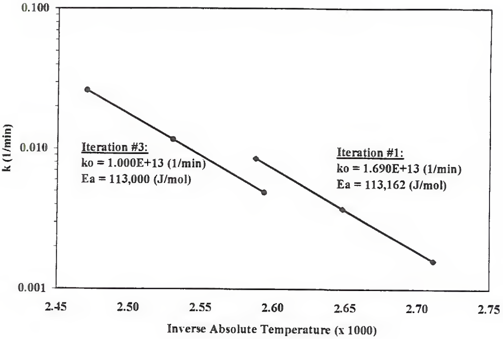


Figure 2-3. Arrhenius plots of first and last iterations of the PEIE method for example #1. Resulting kinetic parameters matched those used to generate the experiment.

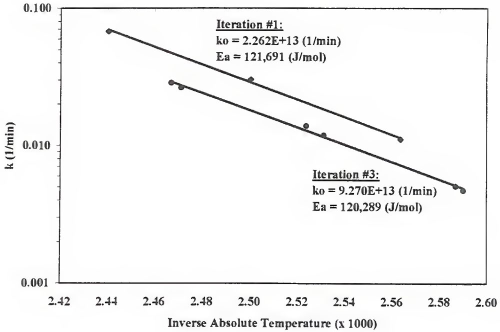


Figure 2-4. Arrhenius plots of first and last iterations of the PEIE method for example #2. Experimental uncertainty was introduced to extent-of-reaction (lethality) data.

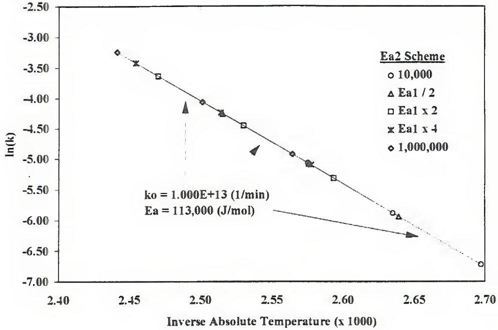


Figure 2-5. Arrhenius plot showing resulting points and regression lines from various $E_a2$ specification schemes for example #1.

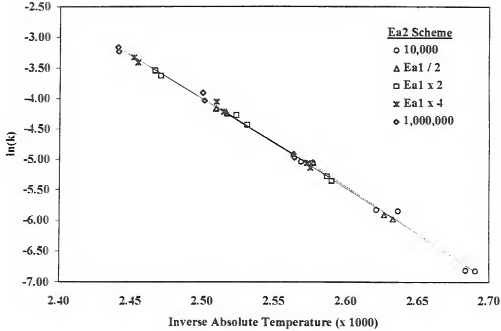


Figure 2-6. Arrhenius plot showing resulting points and regression lines from various $E_a2$ specification schemes for example #2.

# CHAPTER 3
# HEAT TRANSFER SIMULATION FOR THERMAL PROCESS DESIGN

## Introduction

Use of numerical methods for simulating heat transfer for thermal process calculations continues to increase in sympathy with the prevalence and power of modern computers. Numerical methods are favored because, in many cases, analytical solutions do not exist. Numerical methods sacrifice the freedom of ascertaining information at any location and time, by providing data at only specific points in space and time. Efforts to mitigate this limitation spawned the development of numerous algorithms, each offering particular benefits and drawbacks. Thermal process analysis embodies both heat transfer simulation and reaction kinetics. Development of a reliable thermal process analysis tool requires proper selection of a numerical method, and a sound approach for incorporating discrete time-temperature data into lethality (extent-of-reaction) predictions.

Desirable aspects of any numerical method include accuracy, minimal computation time and ease of programming. Programming ease is typically sacrificed for gains in accuracy and/or overall computation time. Methods of sufficient accuracy for thermal process calculations include both explicit and implicit algorithms. Briefly, explicit algorithms are so named because all information required to generate temperature estimates at each time step is explicitly specified by temperature estimates of the previous step. On the other hand, implicit algorithms require information that must be determined concurrently with each temperature estimate. As a result, implicit algorithms are more difficult to program and require greater computational time on a per time step basis (Thibault, 1985). However, implicit methods are inherently stable, allowing greater leaps in time, resulting in fewer steps and overall time savings.

Explicit methods are often criticized because the time step is limited by the laws of thermodynamics. The time step must be small enough to ensure that heat is always considered to flow from hotter to colder regions. However, many texts describe explicit methods in a manner that unnecessarily exaggerates the effect of this limitation (e.g. Lienhard, 1987; Chapman, 1984). This is caused by a seemingly innocuous habit of associating thermal mass with the two-dimensional surface of the object. Both explicit and implicit heat transfer approaches involve estimating energy flows between previously defined points, which are often referred to as nodes. Since nodes are one-dimensional points, volume elements constructed around nodes, provide the corresponding thermal capacity required to construct energy balance equations. The common practice of defining equally spaced nodes results in volume elements whose nodes are on the surface. In the terminology of Clausing (1981), nodes on the surface may be considered, *capacitance surface nodes* (CSN). As demonstrated below, this approach suffers from the fact that the description of the problem is physically inconsistent, because instantaneous temperature changes of the two-dimensional surface are not consistent with actual temperature changes of associated three-dimensional volume elements. This results in a situation where the maximum allowed time step for a stable solution decreases as the surface heat transfer coefficient increases, unnecessarily increasing computational time.

Analyzing such problems using non-capacitance surface nodes (NCSN) results in a physically consistent depiction of the problem and a practical time step restriction. In this case, surface nodes are not considered to have thermal mass. Thus, in a convective environment, heat transfer *through* the surface is analyzed as a combination of resistances to heat transfer involving both convection and conduction. When treated in this manner, the time step is bounded as the convective heat transfer coefficient increases. Although very little evidence of the NCSN approach exists in the literature, it is ideally suited for many thermal processing situations. Researchers who have used the NCSN approach include Bellagha and Chau (1985), Chau and Gaffney (1990) and Silva et al. (1992).

Estimation of the impact of a process on a thermally labile constituent requires integration of the temperature sensitive function (usually the Arrhenius equation) over the temperature history, including space and time. Errors are expected with all numerical methods, because data are available at only certain points in space and time. Since the process of spatial discretization is similar for all finite difference methods, no particular numerical method offers significant advantages over others. Although implicit methods do not restrict time step size, errors related to estimation of extent of reactions during each time interval, grow as the time step increases. Thus, when the goal is extent-of-reaction prediction, an explicit algorithm with a practical time step would be preferable to an implicit method with a similar time step, because the explicit algorithm is both easier to program and executes faster.

The objectives of this work were (1) to compare methods of formulation of explicit methods using both the capacitance and non-capacitance surface node approaches, (2) to demonstrate that both methods offer comparable accuracy, but with greatly different time step restrictions, and (3) to present guidelines for using discrete time-temperature data for approximating extents of reactions (lethality) in systems whose thermal histories are estimated with numerical heat transfer methods.

## Methods

### Overview

Two approaches are commonly used to develop finite difference representations, namely, (1) truncated Taylor expansions and (2) energy balance. The first approach is often described in mathematics and heat transfer texts (e.g. Stoer and Bulirsch, 1993; Ozisik, 1985; Haberman, 1983), and was employed in the earliest applications for thermal processing (Teixeira et al., 1969). The text by Ozisik (1985) provides an excellent presentation on the development of finite difference equations using both difference quotients and energy balances; however, both reflect the CSN case. The second approach offers greater flexibility, because energy balance equations are developed on a volume element basis, allowing location specific variations to be considered. Clausing (1981, p. 170) strongly recommended the second approach by stating,

> . . . if one blindly employs Taylor's expansion or various difference quotients to obtain required difference representations, a physically consistent set [of equations] will probably not result. . . Although it may be possible to obtain a physically consistent set employing Taylor's expansions, it is not always clear how this is done nor how one would be assured of this consistency without making a comparison with the physically derived relationships.

Application of finite difference methods are often described by one or two-dimensional problems in Cartesian coordinates. Chau and Gaffney (1990) demonstrated the spherical case, and Silva et al. (1992) considered infinite geometries. Many practical problems in thermal processing involve finite cylinders. For these reasons application of explicit finite difference methods using the energy balance approach on a finite cylinder was considered. Specifically, CSN and NCSN approaches were developed for an arbitrary conduction heating finite cylinder, subjected to convective boundary conditions. For both cases, the following steps were performed: (1) define node locations and volume element boundaries, (2) perform energy balances on each type of node, and (3) develop equations for determining the maximum allowed time step for stability.

## Nodes and Volume Elements

Spatial descritization is a critical step in finite difference heat-transfer modeling. A node is considered to reside in the central region of a volume element. Each volume element is defined by imaginary boundaries that represent separate control volumes, on which to perform energy balances. Although nodes may be placed arbitrarily, it is convenient to use a standard regimen. The most straightforward method is to separate all nodes by equal distances along the dimensions of the object. This is the typical basis of the CSN approach. Thus for Nr nodes along the radius, and Ny nodes along the half-height, distances, dr and dy, separate nodes along the radius and half-height, respectively. These distances are calculated as follows (see Fig. 3-1):

$$dr = Radius / (Nr - 1) \qquad (3\text{-}1a)$$

$$dy = HalfHeight / (Ny - 1) \qquad (3\text{-}1b)$$

The situation is slightly different for the NCSN case (Fig. 3-2). For Nr nodes along the radius, all *interior* nodes (node 1 through node Nr - 1) are separated by the same distance, dr, while the distance between the last interior node, Nr - 1, and the *surface* node, Nr, is dr/2. This provides 2·Nr - 3 subintervals of length dr/2 along the radius. Thus, distances, dr and dy, between interior nodes along the radius and half-height are given by (see Fig. 3-2 and Fig. 3-4)

$$dr = 2 \cdot Radius / (2 \cdot Nr - 3) \qquad (3\text{-}2a)$$

$$dy = 2 \cdot HalfHeight / (2 \cdot Ny - 3) \qquad (3\text{-}2b)$$

Visual Basic code for discretizing a finite cylinder using the NCSN is shown in Fig. 3-3. Note that several lines of code (Fig. 3-3) have additional code in square brackets (e.g. [dr = Radius / (Nr - 1)]). Substitution of the bracketed code provides the CSN case.

Knowledge of surface temperatures are not required to estimate internal temperatures with the NCSN approach, nor are surface temperatures required for estimation of extent of reactions (lethality). In practice, however, it is worthwhile to develop equations for surface nodes so that they are accounted for in computer code, in case the information is subsequently desired. It is important to remember that for the

NCSN approach, surface nodes are not necessary and are treated differently from interior nodes.

Specification of volume element boundary locations requires consideration of the geometry of the object. Heat flowing vertically in an upright finite cylinder passes through cross-sections of equal area, from the bottom to the top of the cylinder. Heat flowing along the radial direction, however, passes through non-constant areas. Thus, tops and bottoms of volume elements are placed at the mean distance between nodes. Sides of volume elements are placed at the log-mean radius between nodes (Fig. 3-3):

$$r_{lm} = (r_o - r_i) / \ln(r_o / r_i) \qquad (3\text{-}3)$$

where $r_o$, $r_i$, and $r_{lm}$ are outer radius, inner radius and log mean radius, respectively. Since volume elements containing the vertical center-line have a zero inner radius, $r_i$, the boundary may be specified at the mean distance between the first and second radial nodes. It should be noted that for small dr, very little error is introduced by simply using the mean distance between nodes (Lienhard, 1987).

The area of the side of a volume element is equal to the height of the element times the circumference of the circle defined by the radial location of the boundary. The areas of the top and bottom of each node are equivalent. The area of the tops/bottoms are equal to the difference of circular areas defined by the radial locations of the outer and inner side boundaries (Fig. 3-3).

Volumes of elements for both approaches are calculated in a similar manner. Since each node for the CSN approach resides within a volume element, Nr x Ny volume elements are created. In the NCSN case, surface nodes do not reside in volume elements, thus Nr - 1 x Ny - 1 volume elements are created (Fig. 3-3). The volumes of each element may be calculated by multiplying the area of the top boundary by the height of the volume element.

## Energy Balances

Once nodes and volume elements are specified, energy balances may be performed. The first step is to group volume elements according to common geometries and heat-transfer situations. Nine types of volume elements were identified for both the CSN and NCSN nodal arrangements for a finite cylinder (labeled A - I, Fig. 3-1 and Fig. 3-2 and Table 3-1 and Table 3-2). In writing energy balance equations, it is customary to consider heat flowing toward each node through the perpendicular area of the volume element boundary. For both the CSN and NCSN approaches, at least three possibilities exist for heat flow into a volume element. For the CSN case: (1) no heat flow, (2) heat flow via conduction, and (3) convection at the surface. For the NCSN case: (1) no heat flow, (2) heat flow via conduction, and (3) convection at the surface followed by conduction to the node. Other modes of heat transfer are possible, such as radiation, but the three mentioned are of primary importance in thermal processing.

Volume element type C for the CSN approach (Fig. 3-1 and Table 3-2) and the NCSN approach (Fig. 3-2 and Table 3-2), exhibits all three of the respective heat transfer possibilities. Derivation of finite difference equations using energy balances for both approaches are demonstrated below (for complete code see Appendix B):

Capacitance Surface Nodes (CSN)

The node is located on the surface of the finite cylinder. As viewed in Fig. 3-1 and in the diagram for node type C in Table 3-1, convection occurs through the right boundary of the volume element. Conduction occurs through the top and left boundaries, and no heat is transferred from the bottom boundary. The energy balance for this node is (i = Nr, j = 1)

$$\frac{\rho \cdot c_p \cdot dV_{i,j}}{dt}\left(T_{i,j}^{n+1} - T_{i,j}^{n}\right) = \frac{k \cdot rA_{i-1,j}}{dr}\left(T_{i-1,j}^{n} - T_{i,j}^{n}\right) + \frac{k \cdot yA_i}{dy}\left(T_{i,j+1}^{n} - T_{i,j}^{n}\right) \\ + h \cdot rA_{i,j} \cdot \left(T_{\propto}^{n} - T_{i,j}^{n}\right) \tag{3-4}$$

where $\rho$, $c_p$, and k are density, heat capacity, and thermal conductivity of the contents of the finite cylinder, $rA_{i-1,j}$ and $rA_{i,j}$ are the inner and outer side boundary areas of the volume element containing node (i, j), $yA_i$ is the area of the top and bottom of the volume element, $dV_{i,j}$ is the volume of the element containing node (i, j), $T_{i-1,j}^{n}$ and $T_{i,j+1}^{n}$ are previous temperatures of nodes immediately to the right and above node (i, j), $T_{\infty}$ is the ambient temperature, and $T_{i,j}^{n}$ and $T_{i,j}^{n+1}$ are previous and current temperatures of node (i, j). For appropriate nodes, subscripts (i + 1, j) and (i, j - 1) refer to nodes immediately to the left and immediately below node (i, j), respectively.

Solving Eq. 3-4 for $T_{i,j}^{n+1}$ yields

$$T_{i,j}^{n+1} = T_{i,j}^{n} \cdot \left(1 - cL - cT - cR\right) + cL \cdot T_{i-1,j}^{n} + cT \cdot T_{i,j+1}^{n} + cR \cdot T_{\infty}^{n} \tag{3-5}$$

where cL, cT and cR represent coefficients related to heat flow directed toward the node from the left, top and right, respectively (definitions of cL, cR, cT were provided in Table 3-1). The coefficient, cB, was used for volume elements with heat flow from the bottom. The ratio k / ($\rho \cdot c_p$) is thermal diffusivity, denoted by the Greek letter, $\alpha$.

In order for Eq. 3-5 not to violate the laws of thermodynamics (i.e. specifying heat flows from cold to hot regions), the sum of terms within parentheses (1 - cL - cT - cR) must not be negative. Solving this for the time step, dt, provides

$$\mathrm{dt}_{\mathrm{Node\ C}}^{\mathrm{CNS}} \leq \frac{1}{\dfrac{1}{\rho \cdot c_p \cdot dV_{i,j}} \cdot \left( \dfrac{k \cdot rA_{i-1,j}}{dr} + \dfrac{k \cdot yA_{i,j+1}}{dy} + h \cdot rA_{i,j} \right)} . \tag{3-6}$$

Eq. 3-6 shows that as the heat transfer coefficient, h, increases, the time step, dt, must decrease.

## Non-Capacitance Surface Nodes (NCSN)

In this case, the node within volume element type C is at a distance dr/2 from the surface (Fig. 3-2 and Table 3-2). The temperature at the node is affected by conduction through the left and top boundaries, and by convection and conduction from the right boundary. The energy balance is written as follows (i = Nr - 1, j = 1):

$$\begin{aligned} \frac{\rho \cdot c_p \cdot dV_{i,j}}{dt}\left(T_{i,j}^{n+1} - T_{i,j}^{n}\right) = \frac{k \cdot rA_{i-1,j}}{dr}\left(T_{i-1,j}^{n} - T_{i,j}^{n}\right) + \frac{k \cdot yA_{i}}{dy}\left(T_{i,j+1}^{n} - T_{i,j}^{n}\right) \\ + \frac{T_{\infty}^{n} - T_{i,j}^{n}}{\dfrac{1}{h \cdot rA_{i+1,j}} + \dfrac{dr/2}{k \cdot rA_{i,j}}} \end{aligned} \tag{3-7}$$

Solving Eq. 3-7 for $T_{i,j}^{n+1}$ yields

$$T_{i,j}^{n+1} = T_{i,j}^{n} \cdot \left(1 - cL - cT - cR\right) + cL \cdot T_{i-1,j}^{n} + cT \cdot T_{i,j+1}^{n} + cR \cdot T_{\infty}^{n} \tag{3-8}$$

Eq. 3-8 is similar in form to Eq. 3-5, but the coefficient, cR, includes terms related to both convection and conduction. Similarly, the sum of the terms in parentheses must be non-negative. Thus, the maximum time step for the NCSN case is given by

$$dt^{NCNS}_{Node\,C} \leq \frac{1}{\dfrac{1}{\rho \cdot c_p \cdot dV_{i,j}} \cdot \left( \dfrac{k \cdot rA_{i-1,j}}{dr} + \dfrac{k \cdot yA_{i,j+1}}{dy} + \dfrac{1}{\dfrac{1}{h \cdot rA_{i+1,j}} + \dfrac{dr/2}{k \cdot rA_{i,j}}} \right)} . \qquad (3\text{-}9)$$

Aside from describing the physical situation more appropriately, the superiority of the NCSN approach is clearly demonstrated by Eq. 3-9. As the heat transfer coefficient, h, increases, the term representing the resistance to convective heat transfer becomes less significant; this limits the impact of the heat transfer coefficient, h, on the time step, dt. If h is considered to be infinite, as is often done (Shin and Bhowmik, 1990; Teixeira et al., 1969), the time step restriction reduces to that of a specified temperature on the boundary (Type I or Dirichlet boundary condition).

Once interior node temperatures are known, temperatures of surface nodes may be calculated by considering the following heat flux balances:

Side surface ($i = Nr - 1,\ 1 \leq j < Ny - 1$):

$$\frac{k \cdot rA_{i,j}}{dr/2}\left(T^{n+1}_{i,j} - T^{n+1}_{i+1,j}\right) = h \cdot rA_{i+1,j} \cdot \left(T^{n+1}_{i+1,j} - T^{n+1}_{\infty}\right). \qquad (3\text{-}10)$$

Top surface ($1 \leq i < Nr - 1,\ j = Ny - 1$):

$$\frac{k \cdot yA_{i}}{dy/2}\left(T^{n+1}_{i,j} - T^{n+1}_{i,j+1}\right) = h \cdot yA_{i} \cdot \left(T^{n+1}_{i,j+1} - T^{n+1}_{\infty}\right). \qquad (3\text{-}11)$$

Side and top surface temperatures are determined by solving Eq. 3-10 and Eq. 3-11 for, $T_{i+1,j}^{n+1}$ and $T_{i,j+1}^{n+1}$, respectively. Notice that the time step, dt, does not appear in Eq. 3-10 or Eq. 3-11. This demonstrates that surface temperatures are not required to specify internal temperatures, however, surface temperatures may be computed as desired.

Energy balances for all nine types of nodes were performed in a similar manner. Resulting coefficients and equations for the CSN and NCSN approaches were provided in Tables 3-1 and 3-2, respectively. The maximum allowed time step depends on (1) the number of nodes, (2) can dimensions, (3) thermophysical properties of the product, and (4) convective conditions. Thus, a time step no larger than the smallest maximum time step from all nodes should be used. Once the smallest maximum time step is specified, heat transfer simulation may be performed. Before entering the simulation *loop*, node temperatures are set according to a known initial temperature profile. When the simulation loop is entered, the ambient temperature is retrieved, time is advanced by the time step, and the energy balance equations provide temperatures of respective nodes. Figure 3-4 shows an example of programming code for performing these steps.

## Results and Discussion

It may be shown that more complicated algorithms offer smaller degrees of error in nodal temperature estimates due to higher order differential approximations (Press et al., 1992; Clausing, 1981). However, when properly employed, explicit methods have been shown to be sufficiently accurate for the purpose of thermal process calculations (Thibult, 1985). In order for the NCSN approach to offer value in thermal process design, it must provide considerable time savings as well as comparable accuracy to the CSN approach.

Alleviation of the time step restriction afforded by the NCSN approach is demonstrated in Table 3-3. Several standard industrial can sizes were selected for comparison (Miltz, 1992), as well as a plastic can, similar to that studied by Shin and Bhowmik (1990). High and low heat transfer coefficients, h, were chosen to represent external heating and cooling by condensing steam and cooling water, respectively (Walas, 1988). Table 3-3 shows that the maximum time step allowed for the CSN approach was consistently much lower than that for the NCSN approach. Additionally, high h values further restricted the time step for the CSN approach. The fact that the time step was unaltered for the NCSN case demonstrated that the smallest maximum time step was specified by an interior node (for both values of h). Indeed, it is often observed that the volume element containing node type A (Fig. 3-2) provides the smallest maximum time step due to its smaller dimensions.

Most traditional cans have thin metal walls that are often ignored in heat transfer simulations. This is due to their relatively insignificant resistance to heat transfer, as compared to the contents of the can. However, retortable plastic cans are becoming increasingly popular. Shin and Bhowmik (1990) suggested that ignoring the resistance of the plastic can may be inappropriate. They proposed a finite difference method to model this situation using two sets of nodes, one for the can wall and one for the food. The researchers used this approach because they felt that practical node spacing for the food was too course for the thin can wall. Both sets of nodes were assembled according to the CSN approach. Consequently, dimensions of volume elements in the wall became so small, the authors concluded that explicit finite difference methods were impractical, because of the time step restriction. The authors implemented a more complicated implicit method (which is unconditionally stable with regard to time step), and justified its use by stating that, "the time increment in the numerical equations can be large (~10 s)" (Shin and Bhowmik, 1990, p. 170).

Although the Shin and Bhowmik (1990) model generated reasonable solutions, the explicit NCSN approach would have offered a more practical and convenient method of analysis. Instead of going to great lengths to obtain unnecessary temperature distributions in the walls of the can, it is more appropriate to account for the effect of this additional resistance to heat transfer on the thermal response of the food inside the can. In this case, the last term of Eq. 3-7 would take the form

$$\frac{T_{\propto}^{n} - T_{i,j}^{n}}{\dfrac{1}{h \cdot rA_{ow}} + \dfrac{dr_{w}}{k_{w} \cdot rA_{lmw}} + \dfrac{dr / 2}{k \cdot rA_{i,\,j}}} \tag{3-12}$$

where $rA_{ow}$ is the outside area of the wall, $dr_w$ is the thickness of the wall, $k_w$ is the thermal conductivity of the can wall, and $rA_{lmw}$ is the area at the log mean radius of the wall. In most cases, $dr_w$ is relatively small, thus the term containing $dr_w$ would not be expected to significantly alter the maximum allowed time step. As seen in Table 3-3, the maximum time step for the plastic can using the NCSN approach was 103.1 seconds. An absolute comparison of time steps was not possible, because the number of nodes used by Shin and Bhowmik (1990) was not reported. However, an explicit finite difference method with NCSN clearly offered a practical time step. Additionally, with the same number of nodes and time step, the explicit algorithm with NCSN would execute faster than the implicit method used by Shin and Bhowmik (1990).

Accuracy of the CSN and NCSN approaches were compared relative to an analytical solution for heat transfer in a finite cylinder (Fig. 3-5). The contents of the can were assumed to be initially uniform at 0°C. It was assumed that the can was instantaneously exposed to an ambient temperature of 121°C, with a heat transfer coefficient of 5500 W/($m^2 \cdot$°C). The can had a diameter and height of 60.2 and 33 mm, respectively. Figure 3-5 demonstrates that both methods offered comparable accuracy. Additionally, Fig. 3-5 shows that explicit methods offer sufficient accuracy for thermal process calculations. Incidentally, the time required to obtain solutions at 3, 10 and 20 minutes required 4.6, 15.1, and 30.1 seconds, respectively for the CSN approach, and 0.3, 1.2, and 2.3 seconds with the NCSN approach. The computer used was an IBM compatible (Zenon Corp., Industry, CA), equipped with a 90 MHz Pentium processor (Intel Corp., Santa Cruz, CA) and 16 megabytes of RAM. The program was written in Visual Basic (Microsoft Corp. Seattle, WA) running under Windows for Workgroups 3.11 (Microsoft Corp., Seattle, WA).

## Lethality Calculation from Discrete Time-Temperature Data

Prediction of thermal profiles in foods is only half of the task in thermal process design. Thermally labile factors are usually dispersed continuously throughout the volume of the can. Ideally, prediction of the extent of any reaction requires integration of the temperature sensitive kinetic function over the thermal history for the entire volume of the can. An analytical heat transfer solution provides all of the information necessary to accurately perform this task. In the absence of an analytical solution, an appropriate approximation is required.

The nature of spatially discrete data often requires the assumption that the temperature throughout each volume element is uniform and equal to the temperature at the node. However, nodal temperatures are known at only specific points in time. Thermally labile factors are affected continuously through time and therefore, between the instants in times when temperatures are known. Therefore, the question arises, *for each volume element, how should discrete time-temperature data be used to predict the extent of reaction occurring during each time interval?*

For any given time interval, two temperatures are known for each volume element: (1) temperature at the beginning of the interval, $T_{i,j}^{n}$, and (2) the temperature at the end of the interval, $T_{i,j}^{n+1}$. The true path taken between these points is unknown, but is reflected in observed behavior of thermally labile factors. Fig. 3-6 shows several options available for predicting the extent of a particular reaction. Option *a* refers to the possibility of using temperatures at the end of each interval to predict lethality that occurred throughout the interval. Option *b* refers to the possibility of using temperatures at the beginning of each interval. Option *c* refers to the possibility of using the mean of beginning and ending temperatures. Option *d* refers to the possibility of integrating an interpolating function, such as a linear function. Option *e* refers to the possibility of selecting some temperature other than the mean temperature.

Options *a* and *b* are the most straightforward, and are probably used most often. However, most researchers do not report this detail in their methodology. Option *a* was used by Teixeira et al. (1969). It turns out that options *a* and *b* are essentially equivalent, because the same temperature data series is used for lethality calculations, with the exceptions of the first and last data points. Although option *d* would provide the most accurate result, it is probably never used, because it would be difficult to justify the time required to not only approximate appropriate continuous functions, but also to integrate such functions for each volume element and every time step.

Regardless of the actual temperature profile, a linear approximation becomes increasingly accurate as the time step is reduced. Incidentally, errors from all of the options decrease as the time step is reduced; thus, large time steps offered by implicit methods are not beneficial when the goal is extent-of-reaction prediction. Assuming that a time step is chosen such that a linear profile adequately describes the actual profile during each time interval, it is possible to define some temperature along that profile that results in an equivalent extent of reaction as the actual linear profile (see Chapter 2). This is the objective of option *e*. Since temperature affects the rates of many food related reactions in an exponential manner, reaction rates at temperatures above the mean temperature are exponentially faster than those below the mean (Labuza, 1979). Therefore, for typical thermally labile food constituents, it is expected that an equivalent temperature exists in the temperature range above the mean temperature for each time interval.

Many food related reactions are apparently first order with Arrhenius temperature dependence. A first order rate process is defined by

$$\frac{C}{C_o} = \exp\{-k \cdot t\} \tag{3-13}$$

where $C_o$ is the initial concentration of a *reactant*, C is the concentration at time, t, and k is the reaction rate constant (note that the rate constant, k, in Eq. 3-13 is different from thermal conductivity, k, in Eq. 3-4, Eq. 3-6, Eq. 3-7, Eq. 3-9 and Table 3-1 and Table 3-2). The Arrhenius equation describing the temperature dependence of the rate constant is defined by

$$k = k_o \cdot \exp\left\{\frac{E_a}{R \cdot T_{absolute}}\right\} \tag{3-14}$$

where $k_o$ is a pre-exponential factor, $E_a$ is activation energy, R is the ideal gas law constant, and $T_{absolute}$ is the absolute temperature. An equivalent isothermal process may be defined for any dynamic process (see Chapter 2):

$$\frac{C}{C_o} = -k_o \cdot \int_{t_n}^{t_{n+1}} \exp\left\{\frac{-E_a}{R \cdot T(t)}\right\} dt = -k_o \cdot \exp\left\{\frac{-E_a}{R \cdot T_e}\right\} \cdot (t_{n+1} - t_n) \tag{3-15}$$

where $T_e$ is the equivalent isothermal temperature that is represented by option *e* in Fig. 3-6. For a small time step, $(t_{n+1} - t_n)$, the linear temperature change approximation is

$$T_{i.j}^{n+1} = T_{i.j}^{n} + [\text{slope}] \cdot t\Big|_{t_n}^{t_{n+1}} \tag{3-16}$$

The slope in Eq. 3-16 is, of course, the temperature difference divided by the time step. The slope of the linear temperature change represents how rapidly temperatures change, or how dynamic the process is within a particular volume element for each time interval.

The location of the equivalent temperature depends upon (1) the activation energy, $E_a$, of the reactant (2) the slope of the linear temperature change of a given volume element over a given time interval, (3) temperature level, T, and (4) the duration of the time step. Figures 3-7 and 3-8 show how the location of the equivalent isothermal temperature varies according to these parameters. Figure 3-7 is for a high activation energy constituent (500 kJ/mol), such as a bacterial spore population. Figure 3-8 is for a low activation energy constituent (50 kJ/mol), such as a vitamin (Lund, 1980). The horizontal axis represents the slope of the linear temperature change. High values represent highly dynamic thermal situations. Three time steps (1, 10 and 100 seconds) and three beginning temperatures (70, 120 and 150 °C) were considered.

As expected, Fig. 3-7 and Fig. 3-8 show that as temperature changes become less rapid (more constant or isothermal), the equivalent isothermal temperature approaches the mean value. This is intuitive since the average of two equal values is the same value; for $T = T_{i.j}^{n} = T_{i.j}^{n+1}$:

$$\frac{T+T}{2} = \frac{2 \cdot T}{2} = T. \tag{3-17}$$

The smallest time step (1 second) showed the least variation from the mean temperature. The largest time step (100 seconds) showed the largest variation. The higher activation energy value showed more variation from the mean than the lower value. In all cases, initial temperature was not too important.

Simulations can be used to show that for practical food processes, time steps smaller than 10 seconds are required for a linear temperature change approximation to be valid for a majority of the process time. Although simulations of typical retort processes showed that slopes of temperature changes may be as high as 5 °C/second, slopes for most volume elements, for a majority of the process time, are typically less than 1 °C/second. Therefore, Fig. 3-7 and Fig. 3-8 show that for extent-of-reaction predictions it is appropriate to select temperatures at or within 10% above the mean, for each time-interval.

Table 3-1. CSN finite difference formulas (Fig. 3-1).

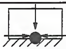
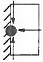
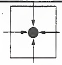
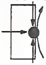
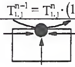
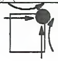
| Type | Diagram | Node(s) | Coefficients |
|---|---|---|---|
| A | | $i = 1$<br>$j = 1$ | $cR = \alpha\cdot\Delta t / dV(i, j) \cdot rA(i, j) / dr$<br>$cT = \alpha\cdot\Delta t / dV(i, j) \cdot yA(i) / dy$ |
| $T^{n+1}_{i,j} = T^{n}_{i,j}\cdot(1 - cR - cT) + cR\cdot T^{n}_{i-1,j} + cT\cdot T^{n}_{i,j+1}$ | | | |
| B |  | $i = 2 .. Nr\text{-}1$<br>$j = 1$ | $cL = \alpha\cdot\Delta t / dV(i, j) \cdot rA(i\text{-}1, j) / dr$<br>$cR = \alpha\cdot\Delta t / dV(i, j) \cdot rA(i, j) / dr$<br>$cT = \alpha\cdot\Delta t / dV(i, j) \cdot yA(i) / dy$ |
| $T^{n+1}_{i,j} = T^{n}_{i,j}\cdot(1 - cL - cR - cT) + cL\cdot T^{n}_{i-1,j} + cR\cdot T^{n}_{i+1,j} + cT\cdot T^{n}_{i,j+1}$ | | | |
| C | | $i = Nr$<br>$j = 1$ | $cL = \alpha\cdot\Delta t / dV(i, j) \cdot rA(i\text{-}1, j) / dr$<br>$cT = \alpha\cdot\Delta t / dV(i, j) \cdot yA(i) / dy$<br>$cR = \alpha\cdot\Delta t / dV(i, j) \cdot h\cdot rA(i, j) / k$ |
| $T^{n-1}_{i,j} = T^{n}_{i,j}\cdot(1 - cL - cT - cR) + cL\cdot T^{n}_{i-1,j} + cT\cdot T^{n}_{i,j+1} + cR\cdot T^{n}_{\infty}$ | | | |
| D |  | $i = 1$<br>$j = 2 .. Ny\text{-}1$ | $cR = \alpha\cdot\Delta t / dV(i, j) \cdot rA(i, j) / dr$<br>$cT = \alpha\cdot\Delta t / dV(i, j) \cdot yA(i) / dy$<br>$cB = \alpha\cdot\Delta t / dV(i, j) \cdot yA(i) / dy$ |
| $T^{n+1}_{i,j} = T^{n}_{i,j}\cdot(1 - cR - cT - cB) + cR\cdot T^{n}_{i+1,j} + cT\cdot T^{n}_{i,j+1} + cB\cdot T^{n}_{i,j-1}$ | | | |
| E |  | $i = 2 .. Nr\text{-}1$<br>$j = 2 .. Ny\text{-}1$ | $cL = \alpha\cdot\Delta t / dV(i, j) \cdot rA(i\text{-}1, j) / dr$<br>$cR = \alpha\cdot\Delta t / dV(i, j) \cdot rA(i, j) / dr$<br>$cT = \alpha\cdot\Delta t / dV(i, j) \cdot yA(i) / dy$<br>$cB = \alpha\cdot\Delta t / dV(i, j) \cdot yA(i) / dy$ |
| $T^{n-1}_{i,j} = T^{n}_{i,j}\cdot(1 - cL - cR - cT - cB) + cL\cdot T^{n}_{i-1,j} + cR\cdot T^{n}_{i-1,j} + cT\cdot T^{n}_{i,j+1} + cB\cdot T^{n}_{i,j-1}$ | | | |
| F |  | $i = Nr$<br>$j = 2 .. Ny\text{-}1$ | $cL = \alpha\cdot\Delta t / dV(i, j) \cdot rA(i\text{-}1, j) / dr$<br>$cT = \alpha\cdot\Delta t / dV(i, j) \cdot yA(i) / dy$<br>$cB = \alpha\cdot\Delta t / dV(i, j) \cdot yA(i) / dy$<br>$cR = \alpha\cdot\Delta t / dV(i, j) \cdot h \cdot rA(i, j) / k$ |
| $T^{n-1}_{i,j} = T^{n}_{i,j}\cdot(1 - cL - cR - cT - cB) + cL\cdot T^{n}_{i-1,j} + cT\cdot T^{n}_{i,j+1} + cB\cdot T^{n}_{i,j-1} + cR\cdot T^{n}_{\infty}$ | | | |
| G | | $i = 1$<br>$j = Ny$ | $cR = \alpha\cdot\Delta t / dV(i, j) \cdot rA(i, j) / dr$<br>$cB = \alpha\cdot\Delta t / dV(i, j) \cdot yA(i) / dy$<br>$cT = \alpha\cdot\Delta t / dV(i, j) \cdot h \cdot yA(i) / k$ |
| $T^{n-1}_{i,j} = T^{n}_{i,j}\cdot(1 - cR - cB - cT) + cR\cdot T^{n}_{i+1,} + cB\cdot T^{n}_{i,j-1} + cT\cdot T^{n}_{\infty}$ | | | |
| H |  | $i = 2 .. Nr\text{-}1$<br>$j = Ny$ | $cL = \alpha\cdot\Delta t / dV(i, j) \cdot rA(i\text{-}1, j) / dr$<br>$cR = \alpha\cdot\Delta t / dV(i, j) \cdot rA(i, j) / dr$<br>$cB = \alpha\cdot\Delta t / dV(i, j) \cdot yA(i) / dy$<br>$cT = \alpha\cdot\Delta t / dV(i, j) \cdot h \cdot yA(i) / k$ |
| $T^{n-1}_{i,j} = T^{n}_{i,j}\cdot(1 - cL - cR - cB - cT) + cL\cdot T^{n}_{i-1,j} + cR\cdot T^{n}_{i+1,j} + cB\cdot T^{n}_{i,j-1} + cT\cdot T^{n}_{\infty}$ | | | |
| I |  | $i = Nr$<br>$j = Ny$ | $cL = \alpha\cdot\Delta t / dV(i, j) \cdot rA(i\text{-}1, j) / dr$<br>$cB = \alpha\cdot\Delta t / dV(i, j) \cdot yA(i) / dy$<br>$cR = \alpha\cdot\Delta t / dV(i, j) \cdot h \cdot rA(i, j) / k$<br>$cT = \alpha\cdot\Delta t / dV(i, j) \cdot h \cdot yA(i) / k$ |
| $T^{n+1}_{i,j} = T^{n}_{i,j}\cdot(1 - cL - cB - cR - cT) + cL\cdot T^{n}_{i-1,j} + cB\cdot T^{n}_{i,j-1} + cR\cdot T^{n}_{\infty} + cT\cdot T^{n}_{\infty}$ | | | |

Table 3-2. NCSN finite difference formulas (Fig. 3-2).

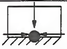
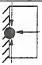
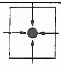
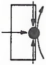
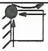
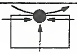
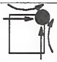
| Type | Diagram | Node(s) | Coefficients |
|---|---|---|---|
| A | | $i = 1$<br>$j = 1$ | $cR = \alpha \cdot \Delta t / dV(i, j) \cdot rA(i, j) / dr$<br>$cT = \alpha \cdot \Delta t / dV(i, j) \cdot yA(i) / dy$ |
| $T^{n-1}_{i,j} = T^{n}_{i,j} \cdot (1 - cR - cT) + cR \cdot T^{n}_{i+1,j} + cT \cdot T^{n}_{i,j+1}$ | | | |
| B |  | $i = 2 .. Nr\text{-}1$<br>$j = 1$ | $cL = \alpha \cdot \Delta t / dV(i, j) \cdot rA(i\text{-}1, j) / dr$<br>$cR = \alpha \cdot \Delta t / dV(i, j) \cdot rA(i, j) / dr$<br>$cT = \alpha \cdot \Delta t / dV(i, j) \cdot yA(i) / dy$ |
| $T^{n+1}_{i,j} = T^{n}_{i,j} \cdot (1 - cL - cR - cT) + cL \cdot T^{n}_{i-1,j} + cR \cdot T^{n}_{i+1,j} + cT \cdot T^{n}_{i,j+1}$ | | | |
| C | | $i = Nr$<br>$j = 1$ | $cL = \alpha \cdot \Delta t / dV(i, j) \cdot rA(i\text{-}1, j) / dr$<br>$cT = \alpha \cdot \Delta t / dV(i, j) \cdot yA(i) / dy$<br>$cR = \alpha \cdot \Delta t / dV(i, j) \cdot h \cdot rA(i, j) / k$ |
| $T^{n+1}_{i,j} = T^{n}_{i,j} \cdot (1 - cL - cT - cR) + cL \cdot T^{n}_{i-1,j} + cT \cdot T^{n}_{i,j-1} + cR \cdot T^{n}_{\infty}$ | | | |
| D |  | $i = 1$<br>$j = 2 .. Ny\text{-}1$ | $cR = \alpha \cdot \Delta t / dV(i, j) \cdot rA(i, j) / dr$<br>$cT = \alpha \cdot \Delta t / dV(i, j) \cdot yA(i) / dy$<br>$cB = \alpha \cdot \Delta t / dV(i, j) \cdot yA(i) / dy$ |
| $T^{n+1}_{i,j} = T^{n}_{i,j} \cdot (1 - cR - cT - cB) + cR \cdot T^{n}_{i+1,j} + cT \cdot T^{n}_{i,j+1} + cB \cdot T^{n}_{i,j-1}$ | | | |
| E |  | $i = 2 .. Nr\text{-}1$<br>$j = 2 .. Ny\text{-}1$ | $cL = \alpha \cdot \Delta t / dV(i, j) \cdot rA(i\text{-}1, j) / dr$<br>$cR = \alpha \cdot \Delta t / dV(i, j) \cdot rA(i, j) / dr$<br>$cT = \alpha \cdot \Delta t / dV(i, j) \cdot yA(i) / dy$<br>$cB = \alpha \cdot \Delta t / dV(i, j) \cdot yA(i) / dy$ |
| $T^{n-1}_{i,j} = T^{n}_{i,j} \cdot (1 - cL - cR - cT - cB) + cL \cdot T^{n}_{i-1,j} + cR \cdot T^{n}_{i+1,j} + cT \cdot T^{n}_{i,j+1} + cB \cdot T^{n}_{i,j-1}$ | | | |
| F |  | $i = Nr$<br>$j = 2 .. Ny\text{-}1$ | $cL = \alpha \cdot \Delta t / dV(i, j) \cdot rA(i\text{-}1, j) / dr$<br>$cT = \alpha \cdot \Delta t / dV(i, j) \cdot yA(i) / dy$<br>$cB = \alpha \cdot \Delta t / dV(i, j) \cdot yA(i) / dy$<br>$cR = \alpha \cdot \Delta t / dV(i, j) \cdot h \cdot rA(i, j) / k$ |
| $T^{n+1}_{i,j} = T^{n}_{i,j} \cdot (1 - cL - cR - cT - cB) + cL \cdot T^{n}_{i-1,j} + cT \cdot T^{n}_{i,j+1} + cB \cdot T^{n}_{i,j-1} + cR \cdot T^{n}_{\infty}$ | | | |
| G |  | $i = 1$<br>$j = Ny$ | $cR = \alpha \cdot \Delta t / dV(i, j) \cdot rA(i, j) / dr$<br>$cB = \alpha \cdot \Delta t / dV(i, j) \cdot yA(i) / dy$<br>$cT = \alpha \cdot \Delta t / dV(i, j) \cdot h \cdot yA(i) / k$ |
| $T^{n-1}_{i,j} = T^{n}_{i,j} \cdot (1 - cR - cB - cT) + cR \cdot T^{n}_{i+1,} + cB \cdot T^{n}_{i,j-1} + cT \cdot T^{n}_{\infty}$ | | | |
| H |  | $i = 2 .. Nr\text{-}1$<br>$j = Ny$ | $cL = \alpha \cdot \Delta t / dV(i, j) \cdot rA(i\text{-}1, j) / dr$<br>$cR = \alpha \cdot \Delta t / dV(i, j) \cdot rA(i, j) / dr$<br>$cB = \alpha \cdot \Delta t / dV(i, j) \cdot yA(i) / dy$<br>$cT = \alpha \cdot \Delta t / dV(i, j) \cdot h \cdot yA(i) / k$ |
| $T^{n-1}_{i,j} = T^{n}_{i,j} \cdot (1 - cL - cR - cB - cT) + cL \cdot T^{n}_{i-1,j} + cR \cdot T^{n}_{i+1,j} + cB \cdot T^{n}_{i,j-1} + cT \cdot T^{n}_{\infty}$ | | | |
| I |  | $i = Nr$<br>$j = Ny$ | $cL = \alpha \cdot \Delta t / dV(i, j) \cdot rA(i\text{-}1, j) / dr$<br>$cB = \alpha \cdot \Delta t / dV(i, j) \cdot yA(i) / dy$<br>$cR = \alpha \cdot \Delta t / dV(i, j) \cdot h \cdot rA(i, j) / k$<br>$cT = \alpha \cdot \Delta t / dV(i, j) \cdot h \cdot yA(i) / k$ |
| $T^{n+1}_{i,j} = T^{n}_{i,j} \cdot (1 - cL - cB - cR - cT) + cL \cdot T^{n}_{i-1,j} + cB \cdot T^{n}_{i,j-1} + cR \cdot T^{n}_{\infty} + cT \cdot T^{n}_{\infty}$ | | | |

Table 3-3. Comparison of maximum time steps allowed for heat transfer simulation via explicit finite difference algorithms. Application of NCSN offers a significant time advantage over CSN.

| Can Size[a,b] | h | # Nodes | Maximum Time Step for Stability | |
|---|---|---|---|---|
| | | | CNS | NCSN |
| | W / ($m^2$·°C) | | (seconds) | (seconds) |
| 54 x 73 mm | 5700 | 8 | 1.70 | 121.20 |
| (202 x 214) | 570 | 8 | 15.67 | 121.20 |
| | 5700 | 12 | 1.08 | 46.45 |
| | 570 | 12 | 9.50 | 46.45 |
| 87 x 116 mm | 5700 | 10 | 3.45 | 233.65 |
| (307 x 409) | 570 | 10 | 32.20 | 233.65 |
| 157 x 178 mm | 5700 | 12 | 2.93 | 360.01 |
| (603 x 700) | 570 | 12 | 27.88 | 360.01 |
| 70 x 73 mm | 5700 | 10 | 1.52 | 103.10 |
| (Plastic) | 570 | 10 | 13.88 | 103.10 |

[a] Values refer to diameter and height, respectively. Values in parentheses follow industrial standards; the first digit designates whole inches, and the remaining digits represent the number of sixteenths of an inch (e.g. 202 = 2 + 2/16 inches).

[b] Dimensions of the plastic can are similar to those reported by Shin & Bhowmik (1990).

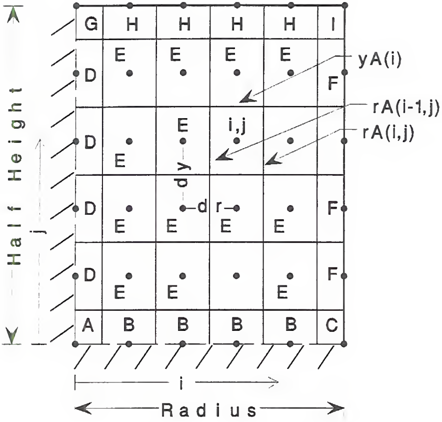


Figure 3-1. Spatial descritization of a finite cylinder for CSN.

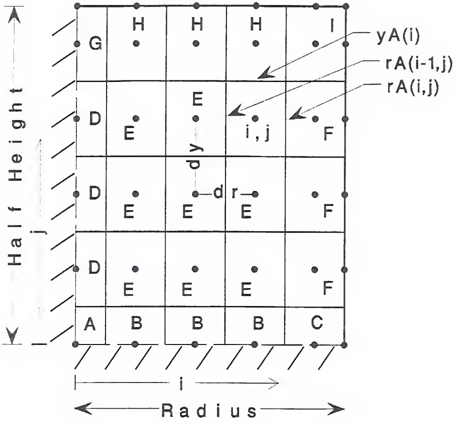


Figure 3-2. Spatial descritization of a finite cylinder for NCSN.

```
ReDim r(Nr), y(Ny), rA(Nr, Ny), yA(Nr), dV(Nr, Ny)
ReDim rALocation(Nr), yALocation(Ny)
Pi = 4 * Atn(1)

  dr = 2 * Radius / (2 * Nr - 3)                CSN-[dr = Radius / (Nr - 1)]
  dy = 2 * HalfHeight / (2 * Ny - 3)            CSN-[dy = HalfHeight / (Ny - 1)]

' NODES IN RADIUS
      r(1) = 0: r(Nr) = Radius
      For i = 2 To Nr - 1
            r(i) = r(i - 1) + dr
      Next i
' NODES IN HALFHEIGHT
      y(1) = 0: y(Ny) = HalfHeight
      For i = 2 To Ny - 1
            y(i) = y(i - 1) + dy
      Next i
' LOCATIONS FOR SIDES OF VOLUME ELEMENTS
      rALocation(1) = r(2) / 2: rALocation(Nr) = Radius
      For i = 2 To Nr - 1
            rALocation(i) = (r(i + 1) - r(i)) / (Log(r(i + 1) / r(i)))
      Next i
' LOCATIONS FOR TOPS AND BOTTOMS OF VOLUME ELEMENTS
      yALocation(1) = y(2) / 2: yALocation(Ny) = HalfHeight
      For j = 2 To Ny - 1
            yALocation(j) = yALocation(j - 1) + dy
      Next j
' VERTICAL AREAS - ⊥ TO RADIAL HEAT FLOW
      For j = 1 To Ny
            For i = 1 To Nr
                  rA(i, j) = 2 * Pi * rALocation(i) * (yALocation(j) - yALocation(j - 1))
            Next i
      Next j
' HORIZONTAL AREAS - ⊥ VERTICAL HEAT FLOW
      yA(1) = Pi * rALocation(1) * rALocation(1)
      yAsum = yA(1)
      For i = 2 To Nr - 1                       CSN-[For i = 2 to Nr]
            yA(i) = Pi * rALocation(i) * rALocation(i) - yAsum
            yAsum = yAsum + yA(i)
      Next i
' VOLUMES
      For j = 1 To Ny - 1                       CSN-[For j = 1 to Ny]
            For i = 1 To Nr - 1                 CSN-[For i = 1 to Nr]
                  dVol(i, j) = yA(i) * (yALocation(j) - yALocation(j - 1))
            Next i
      Next j
```

Figure 3-3. Visual Basic code for discretizing a finite cylinder using NCSN. Code in square brackets may be substituted to yield CSN.

```
Do

TimeNow = TimeNow + TimeStep
AmbientTemperature = GetAmbientTemperature( )
HeatTransferCoefficient = GetHeatTransferCoefficient()

For j = 1 to Ny - 1                          CSN-[For j = 1 to Ny]
   For i = 1 to Nr - 1                       CSN-[For i = 1 to Nr]
      Select Case j
         Case 1
            Select Case i
               Case 1
                  (insert equation for node type A, Table 2 [Table 1])
               Case 2 to Nr - 2              CSN-[Case 2 to Nr - 1]
                  (insert equation for node type B, Table 2 [Table 1])
               Case Nr - 1                   CSN-[Case Nr]
                  (insert equation for node type C, Table 2 [Table 1])
            End Select
         Case 2 to Ny - 2                    CSN-[Case 2 to Ny - 1]
            Select Case i
               Case 1
                  (insert equation for node type D, Table 2 [Table 1])
               Case 2 to Nr - 2              CSN-[Case 2 to Nr - 1]
                  (insert equation for node type E, Table 2 [Table 1])
               Case Nr - 1                   CSN-[Case Nr]
                  (insert equation for node type F, Table 2 [Table 1])
            End Select
         Case Ny - 1                         CSN-[Case Ny]
            Select Case i
               Case 1
                  (insert equation for node type G, Table 2 [Table 1])
               Case 2 to Nr - 2              CSN-[Case 2 to Nr - 1]
                  (insert equation for node type H, Table 2 [Table 1])
               Case Nr - 1                   CSN-[Case Nr]
                  (insert equation for node type I, Table 2 [Table 1])
            End Select
         End Select
   Next i
Next j

For j = 1 to Ny - 1                          CSN-[For j = 1 to Ny]
   For i = 1 to Nr - 1                       CSN-[For i = 1 to Nr]
      OldTemperature(i, j) = NewTemperature(i, j)
   Next i
Next j
```

Figure 3-4. Visual Basic code for main loop of explicit finite difference heat transfer simulation using NCSN. Code in square brackets is for CSN.

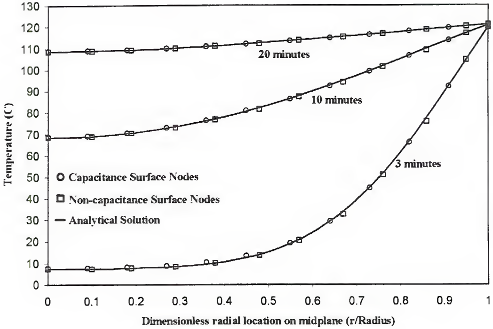


Figure 3-5. Temperature distributions along the radius on the mid-plane of a finite cylinder at 3, 10 and 20 minutes. NCSN offers comparable accuracy to CSN, but with significant time savings.

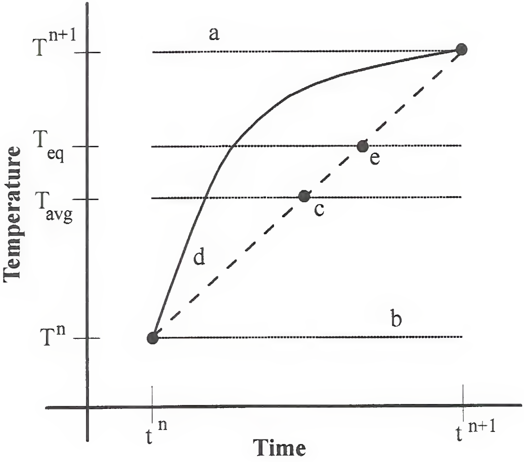


Figure 3-6 Possible temperature selections for calculating extent of reactions (lethality) occurring during each time interval, in each volume element.

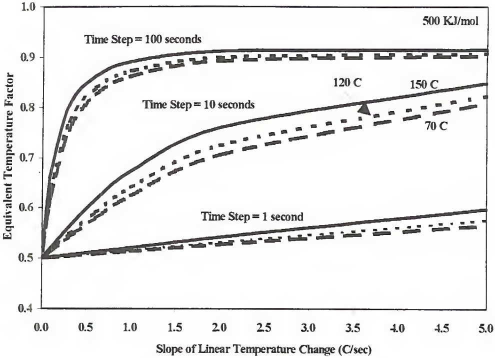


Figure 3-7. Location of isothermal temperature that provides an equivalent extent of reaction (reactant $E_a$ = 500 kJ/mol) as a linearly approximated non-isothermal exposure, between specified initial and final temperatures.

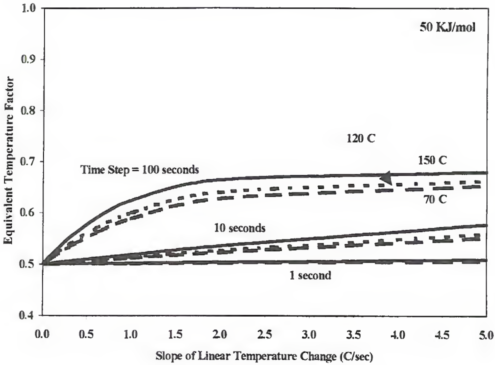


Figure 3-8. Location of isothermal temperature that provides an equivalent extent of reaction (reactant $E_a$ = 50 kJ/mol) as a linearly approximated non-isothermal exposure, between specified initial and final temperatures.

# CHAPTER 4
# KINETIC PARAMETER ESTIMATION IN FOODS THAT DO NOT HEAT UNIFORMLY

## Introduction

Selection of a target microorganism is a critical step in thermal process design. To the process designer, the name of the organism is secondary in importance to the kinetic parameters that reflect its rate of inactivation as a function of temperature. It is well known that the Arrhenius equation is often appropriate for describing the temperature dependence of the reaction rate constant, k, for food constituents and/or microbial inactivation (Cohen and Saguy, 1985). The Arrhenius equation is written as

$$k = k_o \cdot \exp\left\{\frac{-E_a}{R \cdot T_{\text{absolute}}}\right\} \tag{4-1}$$

where the kinetic parameters consist of the pre-exponential factor, $k_o$ [$\text{min}^{-1}$], and the activation energy, $E_a$ [J/mol]. These parameters allow determination of the reaction rate constant, k [$\text{min}^{-1}$], at any temperature, T [Kelvin]. The ideal gas law constant, R, has the value 8.314 J/(mol·Kelvin).

Thermal process design often involves the use of generally accepted kinetic parameters for predicting microbial lethality or constituent losses. Perkins et al. (1975) pointed out that for the case of low acid canned foods, it is customary to use kinetic parameters for *Clostridium botulinum* spores that were determined in the early part of the current century (Esty and Meyer, 1922; Townshend, 1938). These data were derived from traditional isothermal experiments on bacterial spores in aqueous suspensions.

The practice of using *generally accepted* kinetic parameters in thermal process design raises an important question, are the parameters relevant to the product and process at hand? The process engineer must rely on the notion that the parameters represent expected behavior, at least under the conditions studied. Thus, the question addresses the well known fact that bacterial spore inactivation behavior depends upon at least four factors: (1) conditions under which sporulation occurred (Sedlak et al., 1995); (2) conditions of spore recovery (Fernandez et al., 1995; Gonzalez et al., 1995; Aldaton et al., 1974); (3) physical treatment of spores prior to recovery (Mallidis and Drizou, 1991; Gould et al., 1985); and (4) the chemistry of the medium in which the spores are suspended (Ahmed et al., 1995; Juneja et al., 1995; Fernandez et al., 1994; Chai and Liang, 1992; Lynch and Potter, 1988, 1989; Gould, 1985; Perkins et al., 1975). Consequently, the practice of using *generally accepted* kinetic parameters in thermal process design should be reassessed.

Experimental conditions of sporulation and spore recovery may be controlled through adoption of standard methodologies. Exposure of spores to non-thermal, physical injury is not only controllable, but represents an exciting opportunity for product improvement through hybrid process development, geared toward thermal process minimization. Food chemistry, however, is both product and process specific. Obviously, the chemistry of different foods is different. Also, since food chemistry is very sensitive to environmental factors, different processing conditions result in different degrees of chemical change for a given product. Accounting for non-thermal, physical injury of spores and/or variations in food chemistry may permit production of higher quality thermally processed products due to reduced thermal processing requirements. In either case, sufficient impetus exists to seek a better understanding of product and process specific thermal inactivation behavior.

Increasing the complexity of microbial inactivation models is one approach to accounting for non-thermal injury and/or variations in food chemistry (Whiting and Buchanan, 1994). However, practical difficulties arise, since complex models involve a greater number of parameters, which become increasingly difficult to estimate (Sapru et al., 1992). Although complex models may provide important qualitative insight under certain experimental conditions, Welt et al. (1995) found only marginal improvement in lethality predictions in retort processed conduction heating foods, when compared to predictions from the simple first order model with *apparent* parameters. Thus, continued use of the simple first order model for thermal process design is justified.

Another approach for dealing with variations in observed inactivation kinetics is to use simple models (i.e. apparent first order kinetics with Arrhenius temperature dependence), but with product and process specific kinetic parameters. In order to be practical, simple and reliable methods are required for determining kinetic parameters from actual food products subjected to realistic processing conditions.

Two approaches are commonly used to determine thermal kinetic parameters, isothermal and non-isothermal experiments (Lenz and Lund, 1980). The isothermal approach is often criticized for several reasons: (1) limited temperature range from which to select sufficiently different isothermal conditions results in too few conditions to provide good statistical confidence in the parameters (Cohen and Sugey, 1985), (2) unavoidable thermal lags during heat-up and cool-down, (3) many small samples are tedious to prepare, handle and assay, (4) since samples must be small in order to minimize thermal lags, an ideal liquid medium such as a buffer solution, is often used rather than an actual food medium, and (5) even if an actual food product is used, the question of relevance exists, since experimental conditions often differ from processing conditions.

The non-isothermal approach is generally more complicated than the isothermal approach. However, the non-isothermal approach can be applied to uniform and non-uniform heating situations. Uniform heating may be considered to exist in small samples (as in the isothermal approach), well mixed reaction vessels (Welt et al., 1994), or continuous flow systems with turbulent flow (plug flow) conditions. Nonuniform heating occurs with large samples, where time constants of various modes of heat transfer are significant, as with thermally processed canned foods.

For the case of uniform heating, non-isothermal parameter estimation methods overcome the problem of thermal lags, because the entire exposure is considered to affect the result. Non-isothermal methods that account for nonuniform heating alleviate the need for many small samples, since one *sample* is typically an entire product. Additionally, the observed mass average result is a function of the complete thermal profile history of the product/package system.

Nonlinear optimization techniques are often employed for non-isothermal parameter estimation. Chemical engineering literature abounds with approaches and techniques (e.g. Watts, 1994; Lobo and Lobo, 1991; Kim et al., 1990; Espie and Macchietto, 1988; Biegler et al., 1986; Agarwal and Brisk, 1984; Nash, 1977; Marquardt, 1959, 1963). However, most of those reported were applied to well-mixed, uniformly heated systems. Application of nonlinear parameter estimation techniques to nonuniformly heated foods has been demonstrated by Nasri et al. (1993) and Welt et al. (1995). Overall complexity, however, probably limits wide-spread use of non-linear optimization methods.

A non-isothermal approach, applicable to the case of uniform heating, was proposed by Swartzel (1984). However, Maesmus et al. (1995) and Welt et al. (1996) found that a fundamental assertion critical to Swartzel's method was invalid. Welt et al. (1996) proposed a new method, referred to as the method of Paired Equivalent Isothermal Exposures (PEIE). The PEIE method was shown to define kinetic parameters accurately from non-isothermal conditions for the case of uniform heating (Chapter 2).

The purpose of this work was to extend the concept of the PEIE method to nonuniform heating. The goals of this work were (1) to establish the mathematical basis for applying the PEIE method to nonuniform heating, (2) to demonstrate the efficacy of the method by estimating known kinetic parameters from simulated data; and (3) to apply the method to a canned food product inoculated with thermophilic bacterial spores.

## Theory

The following applies to the case of reactants that follow *apparent* first order kinetics with Arrhenius temperature dependence. A first order rate process is defined by

$$\frac{dC}{dt} = -k \cdot C \tag{4-2}$$

where, C is the concentration of a reactant, t is time, and k is the rate constant as defined by Eq. 4-1. Integration of Eq. 4-2 yields

$$\ln(C / C_o) = -k \cdot (t - t_o) \tag{4-3}$$

Thus, extent of a given reaction under isothermal conditions may be determined by substituting Eq. 4-1 into Eq. 4-3:

$$\ln(C / C_o) = -k_o \cdot \exp\left\{\frac{-E_a}{R \cdot T_{\text{absolute}}}\right\} \cdot (t - t_o) \tag{4-4}$$

Under non-isothermal conditions, Eq. 4-2 must be integrated over the entire thermal history, T(t):

$$\ln(C / C_o) = -k_o \cdot \int_0^t \exp\left\{\frac{-E_a}{R \cdot T(t)}\right\} dt \tag{4-5}$$

For brevity, the integral in Eq. 4-5 is referred to as G (Chapter 2):

$$G = \int_0^t \exp\left\{\frac{-E_a}{R \cdot T(t)}\right\} dt \tag{4-6}$$

Clearly, when the thermal history, T(t), is known, a distinct value of G applies to a given value of $E_a$. Additionally, for a given T(t) and $E_a$, an infinite number of equivalent isothermal processes ($t_e$, $T_e$) yield the same value of G via Eq. 4-6:

$$G = t_e \cdot \exp\left\{\frac{-E_a}{R \cdot T_e}\right\} \tag{4-7}$$

Here $t_e$ and $T_e$ represent the equivalent time and isothermal temperature, respectively, that provide the same extent of reaction as the non-isothermal process, for reactants characterized by the particular value of $E_a$. Since the value of G is calculated from Eq. 4-6 using a known thermal history and a known or assumed $E_a$ value, Eq. 4-7, with unknowns, $t_e$ and $T_e$, represents an infinite number of equivalent isothermal processes. This infinite set of equivalent isothermal processes may be observed graphically by plotting a rearranged form of Eq. 4-7:

$$\ln(t_e) = \ln(G) + \frac{E_a}{R \cdot T_e} \tag{4-8}$$

A plot of $\ln(t_e)$ versus $1/T_e$ yields a line with ln(G) as the intercept and $E_a/R$ as the slope. A low value of $E_a$ results in a line with a shallow slope, while a large $E_a$ results in a steep slope. Specification of two reactants, characterized by $E_{a1}$ and $E_{a2}$, yields two curves from Eq. 4-8, for a given non-isothermal process. The point where the curves intersect may be found by solving the two equations simultaneously (Chapter 2). This point of intersection is the isothermal process, ($t_e$, $T_e$), that would yield the same extents of reaction as the non-isothermal process for reactants characterized by the two activation energies, $E_{a1}$ and $E_{a2}$. This point is refereed to as an Equivalent Isothermal Exposure (EIE).

For the case of uniform heating, it was shown in Chapter 2 that this relationship could be exploited to find unknown kinetic parameters from at least one pair of non-isothermal experiments. The method, referred to as the method of Paired Equivalent Isothermal Exposures (PEIE), is depicted in Fig. 4-1. The figure depicts two iterations of the method for two non-isothermal processes (indicated by "Process 1" and "Process 2") for which extents of reaction for a given reactant are known. For the first iteration, two $E_a$ values are arbitrarily selected ($E_a$-1a and $E_a$-1b). For each dynamic thermal exposure considered, the lower $E_a$ value, $E_a$-1b, yields curves with shallower slopes, while the higher value, $E_a$-1a, yields curves with steeper slopes. Points of intersection of respective curves, represent EIE's for these two arbitrarily selected $E_a$ values. An estimation of the *apparent* or *true* $E_a$ value is obtained by combining the EIE specifications ($t_e$, $T_e$) to previously measured extent-of-reaction data. New $E_a$ guesses, $E_a$-2a and $E_a$-2b, are obtained by replacing $E_a$-1a with the calculated $E_a$ estimate and setting $E_a$2b to either a fixed value or some multiple of $E_a$-2a. Schemes for specifying $E_a$-2b were discussed in Chapter 2. The procedure is repeated until $E_a$1 stops changing. Kinetic parameters, $E_a$ and $k_o$, are obtained in a typical manner by constructing an Arrhenius plot from the final equivalent isothermal process specifications and extent-of-reaction data.

The PEIE method generates a set of kinetic parameters for each pair of non-isothermal experiments performed. In the context of the isothermal approach, this is equivalent to drawing a line through each pair of points plotted on an Arrhenius plot (ln(k) versus $1/T_{absolute}$). Thus, for n non-isothermal experiments, n·(n-1)/2, potential sets of kinetic parameters may be defined. For example, for 4 non-isothermal experiments (a, b, c, d), a maximum of 6 sets of kinetic parameters may be expected (one from each pair: a-b, a-c, a-d, b-c, b-d, c-d).

Application of the PEIE method for situations of nonuniform heating was developed for the case of conduction heating canned foods. Complications arise because nonuniform thermal profiles result in nonuniform concentration profiles of reactants within such products. Also, in actual experiments it is often only possible to measure final mass average concentration of reactants (e.g. bacterial spores). Thus, a means to predict transient thermal profiles in a product, and to estimate resulting concentration profiles of a reactant are essential components of the PEIE method for the case of nonuniform heating.

Numerical heat transfer algorithms are often used to obtain transient thermal profiles throughout a product. An in-depth discussion of such methods for the case of a conduction heating finite cylinder was provided in Chapter 3. Briefly, numerical heat transfer algorithms are derived by considering that the product is comprised of discrete volume elements (Fig. 4-2). Accurate temperature estimates are obtained for a specific point, or node, in each volume element at specific points in time. Since volume elements are relatively small, it is assumed that temperatures at the nodes are applicable to the entire volume of respective volume elements. Thus, volume elements are considered to heat uniformly according to non-isothermal processes specified by temperature histories of respective nodes.

In order to apply the PEIE method to nonuniform heating when only final mass average concentration of a reactant is known, additional considerations are required. Initial concentrations of reactants in volume elements are typically equal to the initial concentration of the entire product. Since numerical heat transfer analysis yields a non-isothermal process for each volume element, final concentrations of a reactant in all volume elements may be calculated. Final volume element concentrations may be converted to overall mass average concentration by multiplying each final concentration by the volume of the element (giving the absolute amount of reactant in each volume element), summing these values over all elements, and dividing by the total volume of the object.

Specifically, each volume element (i, j) receives a particular thermal history, $T(t)_{i,j}$. Thus, values of $G_{i,j}$ may be calculated for each volume element, for assumed $E_a$ values. Rearranging Eq. 4-4 and Eq. 4-5 yields

$$\frac{C_{i,j}}{C_o} = \exp\left\{-k_o \cdot G_{i,j}\right\} \tag{4-9}$$

where $C_{i,j}$ is the concentration in volume element (i, j) at any given time. Multiplying each side by the volume of the respective element $\Delta V_{i,j}$, and the known initial concentration, $C_o$, yields

$$N_{i,j} = C_o \cdot \Delta V_{i,j} \cdot \exp\left\{-k_o \cdot G_{i,j}\right\} \tag{4-10}$$

where $N_{i,j}$ is the absolute amount of reactant in volume element (i, j). Summing Eq. 4-10 over all volume elements and dividing by the sum of volume element volumes, yields

$$\frac{\sum N_{i,j}}{\sum \Delta V_{i,j}} = \frac{C_o \cdot \sum\left(\Delta V_{i,j} \cdot \exp\left\{-k_o \cdot G_{i,j}\right\}\right)}{\sum \Delta V_{i,j}} \qquad (4\text{-}11)$$

where the left hand side is the discrete definition of mass average concentration. This provides the basis for comparison with actual experimental results involving final mass average concentrations.

For a particular thermal exposure, and selected value of $E_a$, the pre-exponential factor, $k_o$, is the only unknown in Eq. 4-11. Thus, $k_o$ may be found with any convenient root solving algorithm. Since the value of $k_o$ is typically large, it is convenient to work with the equivalent reference rate constant form of the Arrhenius equation (Eq. 4-1). This form of the Arrhenius equation is derived by substituting a reference temperature, $T_r$, and reaction rate constant at the reference temperature, $k_r$, into Eq. 4-1, and then subtracting the resulting equation from Eq. 4-1. The unwieldy $k_o$ value cancels out and the resulting equation takes the form

$$k = k_r \cdot \exp\left\{\frac{-E_a}{R}\left(\frac{T_r - T}{T \cdot T_r}\right)\right\} \qquad (4\text{-}12)$$

where $k_r$ is the reaction rate constant at temperature, $T_r$. Thus, $k_o$ may be replaced with $k_r$ in Eq. 4-9 through Eq. 4-11. Since food engineers are familiar with the orders of magnitude of rate constants of food constituents at 121.1°C, this temperature serves as a convenient reference temperature. Since the value of $k_{121.1°C}$ must be positive, and its order of magnitude is generally known, bisection may be used to determine its value to a specified level of precision (Press et al., 1992). This additional step is the key to applying the PEIE method to the case of nonuniform heating.

Utilization of the PEIE method for dynamic thermal treatment of samples that heat non-uniformly, involves the following steps, which are illustrated in Fig. 4-3:

1. Record initial concentration, ambient thermal history, and final mass average concentration from at least two dynamic thermal processes that result in statistically different extents of reaction (Processes A and B, Step 1, Fig. 4-3).

2. Arbitrarily select two $E_a$ values. Simulate thermal profiles from actual ambient process conditions. This provides unique thermal histories for all volume elements (Step 2, Fig. 4-2). During simulation, apply the $E_a$ values to accumulate G values according to Eq. 4-7. Thus, for two processes, A & B: $E_{a1}$ and $E_{a2}$ give $G_1$ and $G_2$ for each volume element, i, j, for process A and $G_1$ and $G_2$ for each volume element, i, j, for process B.

3. Considering each process separately, use respective experimentally determined mass average concentrations to find reference rate constants ($k_{r1}$ & $k_{r2}$) that correspond to the two $E_a$ values. This is done by applying a root solving algorithm, such as bisection (Press et al., 1986) on Eq. 4-11.

4. For each experiment, select a representative volume element, (i, j), and estimate the resulting concentration ratio, $C_{i,j} / C_o$, for this volume element, for both sets of kinetic parameters ($E_{a1}$-$k_{r1}$ and $E_{a2}$-$k_{r2}$).

5. Using the two $E_a$ values (from Step 2) and corresponding $G_{i,j}$ values (from Step 3), determine Equivalent Isothermal Exposures (EIE: $t_e$, $T_e$) for the selected volume element (from Step 5), by solving the two equations (form of Eq. 4-7) simultaneously. Using Eq. 4-3, determine respective equivalent isothermal rate constants, $k_T$. Since kinetic parameters, $E_{a1}$-$k_{r1}$, provide $(C_{i,j} / C_o)_1$ and parameters, $E_{a2}$-$k_{r2}$, provide $(C_{i,j} / C_o)_2$, exposure for time, $t_e$, at the equivalent isothermal temperature, $T_e$, would result in the same concentration ratios for the respective $E_a$ values.

6. Using the isothermal rate constants from Step 5 from both processes, calculate new respective $E_a$ values. Thus, $E_{a1}$ applied to processes A and B yields a new $E_{a1}$ value. Use these new $E_a$ values for the next iteration (return to Step 2). However, before repeating, test stopping criteria (e.g. stop when both new $E_a$ values are not more then 1 J/mol different than the old $E_a$ values). Once the stopping criteria are satisfied, take the average of the last two sets of kinetic parameters as *apparent* values for that experimental pair.

When apparent kinetic parameters are defined for a given pair of experiments, select another pair of experimental results and return to Step 2. The PEIE method is performed repeatedly, on all pairs of experimental data, where data from one experiment consists of a recorded ambient thermal history and final mass average concentration of a reactant.

## Methods

### Numerical Heat Transfer

The explicit finite-difference method described in Chapter 3 was used to simulate heat transfer in a finite cylinder. For each time interval during simulation, a temperature value 5% above the mean was used to calculate extent of reaction in each volume element during each time interval. Can dimensions, thermophysical properties of the food, applicable heat transfer coefficients, and parameters related to numerical heat transfer simulation were provided in Table 4-1. Sources and methods for obtaining these data are given below in the section related to actual experiments.

### Experimental

#### Simulated experiments

Simulated experiments were performed to demonstrate the use of the PEIE method for the case of non-uniform heating. A *food constituent* was considered to be characterized by the following Arrhenius kinetic parameters: $k_{121.1°C} = 0.2$ min$^{-1}$, and $E_a = 300$ kJ/mol. The constituent was considered to exist uniformly in a can of conduction heating food at an initial concentration of 1 x $10^6$ units/ml.

Six simulated retort experiments were considered (Table 4-2). During retort come-up and heating phases, heat transfer coefficient, hHeat (Table 4-1), was used. During retort cool-down, hCool (Table 4-1), was used. When the retort temperature reached 30°C, the can was considered to have been placed into an ice bath (hIce, Table 4-1). Simulations were halted when the temperature at the center of the can reached

10°C. Final mass average concentrations were recorded in association with respective thermal treatments (Table 4-2).

The ability of the PEIE method to predict the Arrhenius parameters that were used to generate these *observed* data was tested. For each pair of processes, iterations were stopped when new $E_a$ values (see Step 6 above) changed by less than 1 J/mol. The average of the last $E_a$ and $k_r$ estimates were recorded. Since 6 experiments were considered, a maximum of 15 sets of parameters were expected. Complete code of the PEIE method for the case of conduction heating canned foods is provided in Appendix C.

Actual experiments

Actual experiments were performed with canned pea puree, inoculated with spores of *Bacillus stearothermophilus* (ATCC 7953). Frozen peas were purchased from a local supermarket. The peas were freeze dried and crushed into powder. The powder was placed in home canning jars, sterilized with ca. 50 kGy in a 10 MeV electron beam irradiator (Florida Linear Accelerator, Gainesville, FL), and stored at approximately -20°C prior to use.

Pea puree was prepared by reconstituting the freeze-dried pea powder with distilled water (80% distilled water and 20% powder by weight). The spore suspension was included in the reconstitution formula, and was added after the powder and water were well mixed. The puree had an initial pH of approximately 6.8. Measured initial concentration of spores in the puree was 9 (± 1) x $10^5$ CFU/ml.

Cans were cold filled with 95.0 (± 0.5) grams of inoculated puree. Air pockets were expelled from the puree by lightly tapping each can prior to sealing. Absence of noise when shaking indicated minimal head-space remained in the cans. Each can was uniquely identified with permanent ink, and stored in an ice bath for at least 48 hours prior to use. Thermocouples located at the center of several cans, indicated that the center reached 0°C within 24 hours. Thus, initial temperature profiles of all samples were considered to be uniform at 0°C.

During thermal treatments, each sample experienced four ambient conditions. Each condition was reflected in the heat transfer coefficients applied during simulation (Table 1). The four conditions were: (1) after removal from the ice bath and before application of steam in the retort (hAir), (2) during steam heating in the retort (hHeat), (3) during water cooling in the retort (hCool), and (4) placement in an ice bath (hIce), after removal from the retort. A reasonable value of hHeat was assumed since a precise value is not required when it is known to be very high. Values of hAir, hCool, and hIce were inferred by measuring the thermal response of a solid aluminum cylinder to conditions similar to those experienced during treatment. Dimensions of the cylinder were similar to those of the actual cans (Table 4-2). Since the thermal conductivity of aluminum is very high and the cans were relatively small, the temperature of the object was considered to be uniform. Thus, a lumped parameter heat transfer solution provided the means to obtain values of the heat transfer coefficient. The lumped parameter solution has the form (Lienhard, 1987)

$$\frac{T - T_{\alpha}}{T_i - T_{\infty}} = \exp\left\{\frac{h \cdot A}{\rho \cdot c_p \cdot V} \cdot t\right\}, \tag{4-13}$$

where A and V are the area and volume of the aluminum cylinder, $\rho$ and $c_p$ are the density and specific heat of aluminum (Perry and Green, 1984), $T_{\alpha}$ and $T_i$ are the ambient and initial temperature of the aluminum cylinder and T is the temperature of the aluminum cylinder at any time, t. Equation 4-13 shows that the heat transfer coefficient, h, may be inferred from the slope of a semi-log plot of accomplished temperature versus time.

Cans were exposed in duplicate to different retort processes. In general, three approximate set-point temperatures were used (104.4, 112.8, and 120.6°C) in conjunction with various heating times. Ambient temperatures in the retort were recorded with at least three thermocouples in proximity to the cans. Recorded ambient temperature histories and resulting survivor ratios are shown in Fig. 4-4 and Fig. 4-5. Details of microbial enumerations were provided in Appendix D. Fifteen different processes were considered; the processes are numbered in the figures. Processes 4 and 6 (Fig. 4-1) were not plotted because they were similar to processes 3 and 5, respectively. Survival ratios for processes 4 and 6 were 0.272 and 0.068, respectively. Thermocouples were calibrated to a mercury-in-glass (MIG) thermometer installed on the retort. The pilot scale vertical still-cook retort was supplied with 0.17 MPa (25 psig) steam in the pilot plant of the Department of Food Science and Human Nutrition at the University of Florida (Gainesville, FL).

All samples were kept in an ice bath after treatment, and were assayed for surviving spores within 48 hours. The contents of each can were placed into a sterile 250 ml beaker and mixed well for several minutes by hand. The puree was sampled in triplicate by transferring ca. 0.5 grams of puree into respective test tubes with sterile, disposable 10 ml pipettes. Samples were weighed on a digital scale, and immediately diluted by a factor of 10 (by weight) with sterile 0.15 M potassium phosphate buffer (pH 7.0). Serial dilutions were subsequently performed. Four 50 μl aliquots of respective dilutions were placed on petri dishes containing non-selective media. The drops were dried on the surface of the media in a 55°C incubator. The plates were arranged in groups of four and covered lightly with aluminum foil, inverted, and incubated at 55°C for 24 hours. Colonies were manually enumerated.

The recovery medium consisted of 8 grams of BBL nutrient broth (Fisher Scientific, Pittsburgh, PA) and 20 grams of Difco Bacto Agar (Fisher Scientific, Pittsburgh, PA) per liter of distilled water. The medium was autoclave sterilized, and poured into sterile petri dishes (Fisher Scientific, Pittsburgh, PA). After cooling and hardening, the plates were stored in a refrigerator. All plates were used within two weeks of preparation.

## Results and Discussion

The objective of the simulated experiments was to determine whether the PEIE method could identify Arrhenius parameters that were used to generate the *observed* data (Table 4-2). As mentioned above, the parameters used were $k_{121.1°C} = 0.2$ min$^{-1}$

and $E_a$ = 300 kJ/mol. Since 6 experiments were considered, 15 experimental pairs, and thus, a maximum of 15 sets of parameters were expected. Table 4-3 summarizes results from all process pairs. Although some scatter was evident, the PEIE method was capable of defining the original parameters to a level of accuracy consistent with the *observed* data. As shown in Table 4-2, extent-of-reaction data were rounded from 15 (double precision) to 6 decimal places. Statistical analysis of Table 3 yielded $k_r$ = 0.199997 (± 0.000003) $min^{-1}$ and $E_a$ = 299,998 (± 2) J/mol (intervals were determined at the 95% confidence level). Clearly, the PEIE method was successful in identifying the original kinetic parameters to an acceptable level of precision.

Results from experiments on canned pea puree inoculated with spores of *B. stearothermophilus* are shown in Fig. 4-6 and Fig. 4-7. Since 14 experiments were considered, a maximum of 91 experimental pairs, and thus, 91 sets of parameters were possible. However, Fig. 4-6 and Fig. 4-7 show 51 sets. Reasons for the apparent loss of data were (1) experimental uncertainty, (2) inactivation model simplification, and (3) conditions that contributed to loss in numerical precision. These factors are discussed in detail below. Appendix G provides tabulated values and the origin of all data in Fig. 4-6 and Fig. 4-7.

Experimental uncertainty and model simplification have similar effects on kinetics analysis. Ideally (for irreversible, first order kinetics), as a thermal process becomes more severe, the number of surviving spores decreases. In general, this is observed, however, when comparing survivors from comparable processes, the more severe process may yield a greater *mean* survivor level. This may be the result of a more complex inactivation process such as activation of dormant spores (Sapru et al., 1992), or it may simply be the result of overlapping envelopes of experimental uncertainty. In either case, if this situation arises when a simple inactivation model is assumed (i.e. Eq. 4-4), a negative *apparent* activation energy results, which is not possible. In this experiment, spores were not thermally activated prior to use, and neither are they in practical situations. Thus, thermal activation of dormant spores was a probable factor. Additionally, microbial enumeration techniques are notoriously uncertain. Therefore, it is probable that both factors influenced the results. The PEIE method computer code in Appendix C was designed to simply bypass process pairs that generated negative $E_a$ values during execution. This accounted for 6 of 91 possible data points.

Each iteration of the PEIE method involves calculation of activation energy estimates from equivalent isothermal data. Therefore, the PEIE method is restricted in a manner similar to the traditional isothermal kinetics approach. In order to construct an Arrhenius plot from traditional isothermal experiments, reaction rates from at least two temperatures must be known. Thus, the PEIE method suffers when two dynamic thermal processes yield similar equivalent temperatures, $T_e$. This problem was mitigated by two approaches. The first was to select the volume element that provided the greatest difference in equivalent temperatures during each iteration (see Step 5, above). The second was to consider only pairs of experimental processes that had significantly different thermal treatments. Thus, processes associated with one retort set-point temperature were analyzed only with processes associated with different retort set-point temperatures. For example, process 1 (Fig. 4-4) was not paired with processes 2 through 8, but was paired with processes 9 through 14 (Fig. 4-5). This accounted for 34 of 91 possible data points.

As previously mentioned, the initial pH was approximately 6.8. Final mass average pH tended to decrease with increasing process severity, and ranged to as low as approximately 5.8. Histograms of observed $\ln(k_{121.1°C})$ and $E_a$ values showed normal distribution behavior. The observed log-mean $k_{121.1°C}$ value was 0.26 $min^{-1}$, with a 95% confidence interval for the population of 0.23 to 0.30 $min^{-1}$. The observed mean $E_a$ value was 250 kJ/mol, with a 95% confidence interval for the population of ± 15 kJ/mol.

Fundamental differences between the PEIE method and traditional kinetic parameter estimation methods may preclude direct comparisons of estimated parameters. The following discussion will focus on the case of uniformly heating samples, since that is the typical premise of traditional isothermal kinetics experiments. Although microbial survivor curves often possess shoulders during initial phases of inactivation (Fig. 4-8), it is common practice to discard the shoulder, and obtain kinetic parameters from the straight portion of the curve. This is done by thermally activating spores prior to lethal exposure and/or by performing a linear regression on data only in the straight portion of the curve. As shown in Fig. 4-8, use of parameters derived from the straight portion of survivor curves will constantly over predict lethality to an unknown extent when shoulders are present during lethality.

As implemented, the PEIE method only considers irreversible first-order inactivation. Additionally, the PEIE method operates by connecting two points on a survivor curve (Fig. 4-8). Thus, in the absence of a shoulder on the actual survivor curve, the PEIE method would provide results similar to the traditional kinetic parameter estimation approach. When a shoulder does exist, the parameters obtained with the PEIE method will differ from those of the traditional approach. Reed et al. (1951) provided an excellent example of this difference. Amongst experiments involving several bacterial spores and suspending media, they performed experiments with pea puree inoculated with the same strain of spores used in this study. The authors reported the presence of shoulders in survivor curves, and reported results in the following two ways, (1) total time to achieve 99.99% reduction in spore count (four log-cycle reduction) at 121.1°C, and (2) time required for the straight line portion to traverse four log-cycles at 121.1°C. They reported 36 and 25.4 minutes, respectively, for the two methods; each with z values of 11.1°C. The times on a per log-cycle basis are the definition of $D_o$ values (time required to achieve a one log-cycle reduction in concentration. The $D_o$ value is related to the reaction rate constant, $k_{121.1°C}$, via

$$D_o = \frac{\ln(10)}{k_{121.1}} \tag{4-14}$$

The method used by Reed et al. (1951) was analogous to the result provided by the PEIE method. On a $D_0$ and z value basis, the values reported here ($D_0 = 8.9$ min; $z = 11.4°C$) agreed very well with data from Reed et al. (1951) ($D_0 = 9.0$ min; $z = 11.1°C$).

The PEIE method was employed to understand thermal inactivation behavior of a particular spore population, in a specific product, subjected to specific process conditions. In this context, the PEIE method may be used to quantify the impact of ingredient and/or process modifications, as well as non-thermal pre-treatments (e.g. irradiation), on the thermal response of food constituents.

Table 4-1. Parameters used for numerical heat transfer simulation of retort processed canned peas.

| | | | |
|---|---|---|---|
| Thermophysical properties: | | | |
| Thermal conductivity | k | 0.682 | W/(m·°C) |
| Density | $\rho$ | 958 | Kg/m$^3$ |
| Heat capacity | $c_p$ | 4,350 | J/(Kg·°C) |
| Heat Transfer Coefficients: | | | |
| Ambient in pilot plant | hAir | 1 | W/(m$^2$·°C) |
| Steam in retort | hHeat | 5,500 | W/(m$^2$·°C) |
| Cooling water in retort | hCool | 90 | W/(m$^2$·°C) |
| Ice bath | hIce | 60 | W/(m$^2$·°C) |
| Can Dimensions: | | | |
| Diameter | D | 0.0602 | m |
| Height | H | 0.0348 | m |
| Nodes: | | | |
| In radius | Nr | 12 | |
| In half-height | Ny | 12 | |

Table 4-2 Hypothetical retort process conditions and simulated mass average concentration of a reactant ($E_a = 300$ kJ/mol, $k_{121.1°C} = 0.2$ min$^{-1}$).

| Process | Retort Temp<br><br>(°C) | Retort<br>Come-up<br>(min) | Retort<br>Heating<br>(min) | Retort<br>Cool Down<br>(min) | Mass Average<br>Concentration<br>($C / C_o$) |
|---|---|---|---|---|---|
| A | 110 | 5 | 5 | 5 | 0.993233 |
| B | 110 | 5 | 95 | 5 | 0.322360 |
| C | 117 | 7 | 11 | 9 | 0.839512 |
| D | 117 | 7 | 19 | 9 | 0.593906 |
| E | 125 | 8 | 4 | 10 | 0.846742 |
| F | 125 | 8 | 14 | 10 | 0.282007 |

Table 4-3. Pair wise PEIE method results for simulated experimental data.

| Process Pair | | $E_a$ | $k_r$ - 121.1 °C |
|---|---|---|---|
| | | (J/mol) | (min$^{-1}$) |
| A | B | 299,985.28 | 0.1999736 |
| A | C | 299,989.68 | 0.1999851 |
| A | D | 299,991.38 | 0.1999897 |
| A | E | 299,993.54 | 0.1999954 |
| A | F | 299,994.01 | 0.1999966 |
| B | C | 299,999.41 | 0.1999988 |
| B | D | 299,999.57 | 0.1999992 |
| B | E | 299,999.65 | 0.1999994 |
| B | F | 299,999.83 | 0.1999998 |
| C | D | 299,999.71 | 0.1999995 |
| C | E | 299,999.76 | 0.1999993 |
| C | F | 300,000.07 | 0.2000001 |
| D | E | 299,999.70 | 0.1999996 |
| D | F | 300,000.04 | 0.1999999 |
| E | F | 300,004.63 | 0.2000029 |

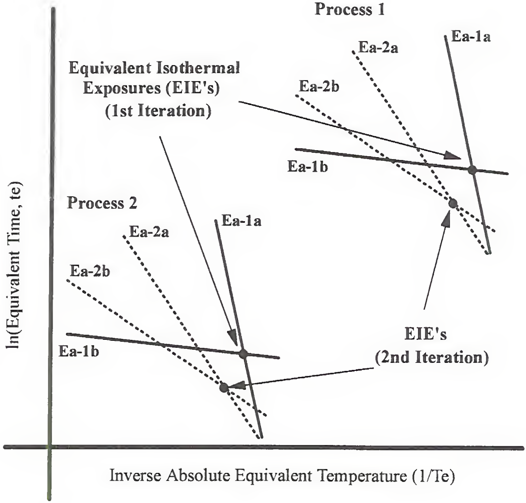


Figure 4-1. Graphical depiction of two iterations of the PEIE method.

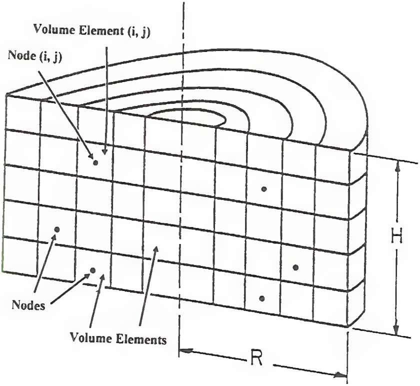


Figure 4-2. Discrete volume elements specified to simulate heat transfer in a finite cylinder.

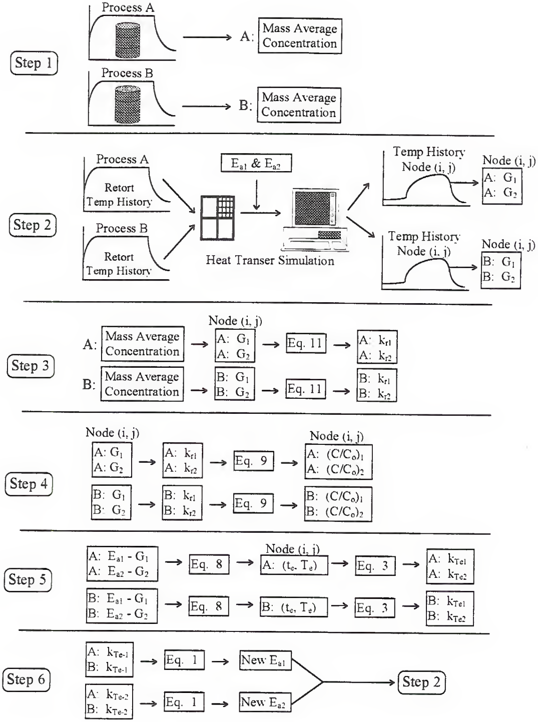


Figure 4-3. Steps for executing the PEIE method for the case of non-uniform heating.

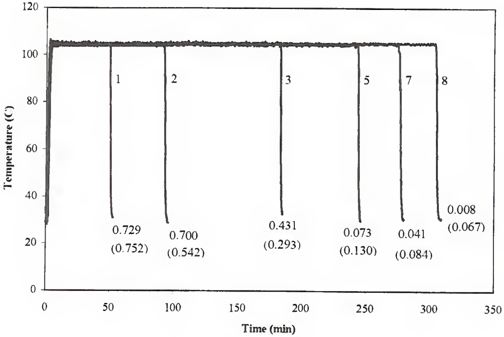


Figure 4-4. Non-isothermal retort treatments applied to cans of pea puree inoculated with spores of *B. stearothermophilus*. Numbers after each process are experimental survivor ratios. Survival ratios in parentheses were predicted from resulting kinetic parameter estimates.

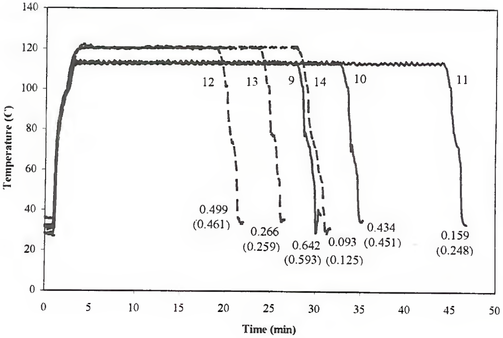


Figure 4-5. Non-isothermal retort treatments applied to cans of pea puree inoculated with spores of *B. stearothermophilus*. Numbers after each process are experimental survivor ratios. Survival ratios in parentheses were predicted from resulting kinetic parameter estimates.

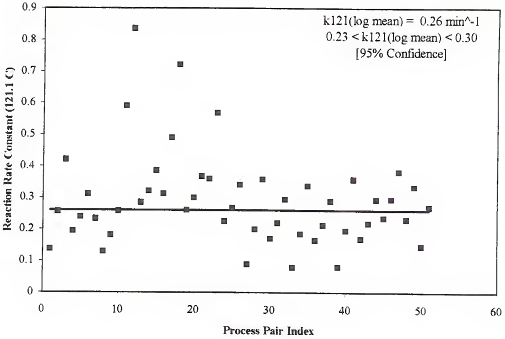


Figure 4-6. Reaction rate constant, k, at 121.1 °C for inactivation of spores of *B. stearothermophilus* in canned pea puree as determined with the PEIE method.

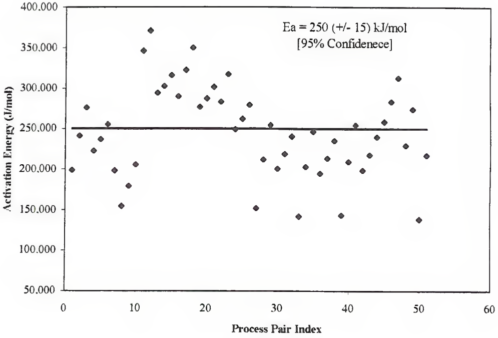


Figure 4-7 Activation energy, $E_a$, values for inactivation of spores of *B. stearothermophilus* in canned pea puree as determined with the PEIE method.

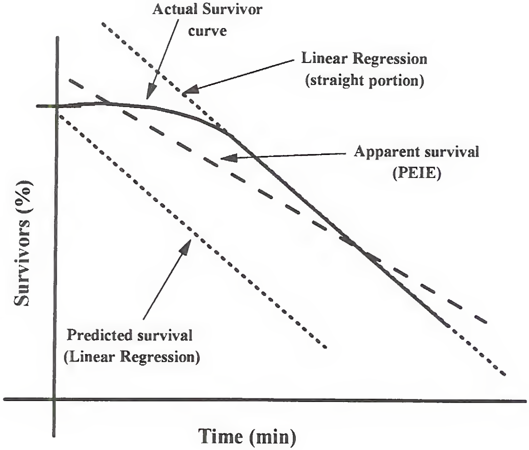


Figure 4-8. Traditional isothermal kinetics typically uses the straight portion of survivor curves. The PEIE method assumes a straight line between two points on the actual survivor curve.

# CHAPTER 5
# EXPLOITING TREATMENTS THAT SENSITIZE BACTERIAL SPORES TO HEAT IN THERMAL PROCESS DESIGN

## Introduction

The goal of thermal processing is inactivation of microorganisms in food products. The goal of thermal process *design* is to insure sufficient microbial inactivation with minimal collateral degradation to product quality. Unfortunately, traditional thermal processing methods offer limited opportunities to reduce process severity, which limits opportunities to enhance product quality.

Kinetic parameters that describe the rate of microbial inactivation as a function of temperature, are central to thermal process design. It is well known that the Arrhenius equation is often appropriate for describing the temperature dependence of the reaction rate constant, k, for food constituents and/or microbial inactivation (Teixeira, 1992). The Arrhenius equation may be written as

$$k = k_r \cdot \exp\left\{\frac{-E_a}{R} \cdot \left(\frac{T_r - T}{T_r \cdot T}\right)\right\} \qquad (5\text{-}1)$$

where the kinetic parameters consist of a reference rate constant, $k_r$ [min$^{-1}$] which is the rate of inactivation at reference temperature, $T_r$ [Kelvin], and the activation energy, $E_a$ [J/mol]. These parameters allow determination of the reaction rate constant, k [min$^{-1}$], at any other temperature, T [Kelvin]. The ideal gas law constant, R, has the value 8.314 J/(mol·Kelvin).

Although it is now generally known that microbial inactivation kinetic parameters are functions of various external factors, the thermal processing industry continues to treat such parameters as immutable (Perkins et al., 1975). Examples of external factors that influence inactivation kinetics include: (1) conditions during sporulation, (2) conditions of spore recovery, (3) physical treatment of spores, and (4) the particular chemistry of the suspending medium. Methods of kinetic parameter estimation were proposed in Chapter 2 and Chapter 4 which were referred to as the method of Paired Equivalent Isothermal Exposures (PEIE). The PEIE method was shown to allow quantification of product and process specific influences on microbial inactivation kinetic parameters in products that do and do *not* heat uniformly (Chapters 2 and 4, respectively). One promising application of such methods is quantification of the impact of additives and/or non-isothermal pretreatments that enhance sensitivity of bacterial spores to heat.

Comprehensive efforts to identify chemical substances that enhance thermal sensitivity of bacterial spores have been made, but with limited success (Michener et al., 1959). Conversely, relatively mild doses of ionizing radiation (< 10 kGy) have been shown to enhance thermal sensitivity of bacterial spores (Shamsuzzaman and Lucht, 1993; Shamsuzzaman et al., 1990; Shamsuzzaman, 1988; Gombas and Gomez, 1978; Stegeman, 1976; Licciardello and Nickerson, 1962, 1963; Kemp et al. 1959, 1960; Kempe et al., 1957, 1958; Kempe, 1955). Regrettably, commercial development of food irradiation technology continues to be stymied by the unscientifically based act of the United States Congress, which forced the Food and Drug Administration (FDA) to view many practical applications of ionizing radiation to foods as a *food additive*. Since methods of assuring safety and efficacy of typical chemical food additives are inappropriate for assessing *foods* exposed to ionizing radiation, the food additive review process is cumbersome and unduly challenging. However, progressive legislative changes regarding food irradiation technologies will probably occur for several reasons, including: (1) extensive research has shown that irradiated foods are wholesome (Thayer, 1994; Diehl, 1990; Skala et al., 1987), (2) growing concerns of food-borne illnesses which are easily controllable via irradiation (Lenius, 1995), (3) realization by the food industry that a significant percentage of consumers will purchase irradiated products (Pohlman et al., 1994; Pszcozola, 1993; Bruhn and Noell, 1987), and (4) recent food processing related innovations in irradiation technology (e.g. Welt and Barnett, 1995; Diehl, 1990).

Three possibilities exist for combining radiation and heat treatments: (1) heat prior to radiation, (2) heat and radiation applied together, and (3) radiation prior to heat. It has been shown that application of heat prior to radiation offers no significant benefit than either process alone (Gould, 1985; Gombas and Gomez, 1978; Kempe, 1955). The case of simultaneous applications of heat and radiation has been termed *Thermoradiation*, and has been extensively studied (Thayer et al., 1991; Schaffner et al., 1989; Fisher and Pflug, 1977; Pallas and Hamdy, 1976; Emborg, 1974; Sivinski et al., 1972; Grecz et al., 1971; Licciardello, 1964). Although results were favorable, specialized commercial scale equipment is not currently available and no practical method exists for predicting processing results. The case of irradiation followed by heat has also shown favorable results and is reviewed below. Hybrid processes involving irradiation followed by thermal treatment probably represents the most immediately viable commercial option due to equipment availability and respective processing know-how.

Sincere research of irradiation followed by thermal treatment began in the middle of the twentieth century. Early work by Kempe (1955) demonstrated considerable enhancement in thermal sensitivity of *Clostridium botulinum* (62A and 213B) spores in pH 7.0 phosphate buffer, pH 6.7 nutrient broth, and pH 7.0 10% gelatin. Kempe et al. (1957, 1958) performed sterilization assessment studies on cooked and raw ground beef in no. 1 picnic tin cans, inoculated with *C. botulinum* 213B spores. For each dose studied, they determined thermal process requirements for sterilization by recording the presence or absence of swelling in incubated cans after treatment. They found that a dose of approximately 10 kGy provided a 4 to 5 fold reduction in $F_o$ value (minutes of thermal processing at 121.1 °C) for both cooked and raw ground beef. Kempe et al. (1959, 1960) continued the research theme by performing similar experiments on *C. sporogenes* PA 3679 spores in cooked ground beef, and *C. botulinum* 213B and *C. sporogenes* PA 3679 spores in canned green peas in brine. The authors reported qualitatively similar results to those previously reported.

Licciardello and Nickerson (1962, 1963) studied effects of oxygen and pH on degree of thermal sensitization achieved with *C. sporogenes* PA 3679 and *Bacillus* *subtilis* ATCC 6633 spores, suspended in pH 7.0 phosphate buffer, nutrient broth and ham puree. Thermal treatments were performed with suspensions in melting-point glass capillary tubes. Overall, they found favorable results with the combined treatments, and confirmed the notion that external factors such as oxygen, pH and suspending medium chemistry affect the thermal response of bacterial spores. Deasy et al. (1970) studied the influence of phenolic bactericides on radiation enhanced thermal sensitivity of *B. subtilis* NCTC 8236 spores in aqueous suspensions in the pH range of 6 to 8. Stegeman et al. (1976) reported evidence to suggest a possible mechanism of radiation enhanced thermal sensitization with experiments performed on spores of *B. subtilis* ATCC 6633 and *B. stearothermophilus* NCA 1518.

Gombas and Gomez (1978) worked with spores of *C. perfringens* type A (NCTC 8798) in pH 7.0 phosphate buffer. They found no benefit by applying heat prior to radiation, but found significant radiation enhanced thermal sensitization. Additionally, they found that the sensitizing effect was stable for at least 28 hours following irradiation. Shamsuzzaman (1988) confirmed the persistence of radiation enhanced thermal sensitization with spores of *C. sporogenes* PA 3679 in pH 7.0 phosphate buffer and nutrient broth. Shamsuzzaman et al. (1990) demonstrated persistence of the sensitizing effect for at least 14 days with *C. sporogenes* PA 3679 spores in pH 7.0 phosphate buffer, distilled water and homogenized commercially sterile canned foods. Shamsuzzaman and Lucht (1993) studied radiation enhanced thermal sensitization of *C. sporogenes* PA 3679 spores in non-aqueous media including, vegetable oils, animal fats, synthetic triglycerides, glycerol and phosphate buffer. They confirmed sensitization in non-aqueous media, but noted different degrees of sensitization from the different media.

A plethora of evidence suggests that relatively mild doses of ionizing radiation can significantly reduce thermal processing requirements. The PEIE method is ideally suited for quantifying the impact of pre-irradiation on microbial inactivation kinetic parameters in actual food products subjected to realistic processing conditions. The purpose of this work was to employ the PEIE method to determine thermal inactivation kinetic parameters for bacterial spores in an actual food product, pre-treated with mild doses of ionizing radiation.

## Methods

Pea puree was used as the food medium. Frozen peas were purchased from a local supermarket. The peas were freeze dried and crushed into powder. The powder was placed in home canning jars, radiation sterilized with approximately 50 kGy from a 10 MeV linear accelerator (Florida Linear Accelerator Research Center, Gainesville, FL), and stored at approximately -20°C prior to use. The puree was prepared by reconstituting the freeze-dried pea powder with distilled water (80% distilled water and 20% powder by weight). A buffered aqueous suspension of *Bacillus stearothermophilus* (ATCC 7953) spores was included in the reconstitution formula, and was added after the powder and water were well mixed. The puree had an initial pH of approximately 6.8. Initial concentration of spores in untreated puree was 9 (± 1) x $10^5$ CFU/ml (see Appendix D).

Aluminum cans with pull-top lids were used. The cans had an inside diameter and height of 60.2 and 33.0 mm, respectively (Table 5-1). The cans were cold filled with 95.0 (± 0.5) grams of inoculated puree. Air pockets were expelled from the puree by lightly tapping each can prior to sealing. Absence of noise when shaking indicated that minimal head-space remained in the cans. Each can was uniquely identified with permanent ink, and stored in an ice bath for at least 48 hours prior to use. Thermocouples located at the center of several cans, indicated that the center reached 0°C within 24 hours. Thus, initial temperature profiles of all samples were considered to be uniform at 0°C.

Samples (cans) were irradiated with bremsstrahlung radiation (x-rays) at the Florida Linear Accelerator (Gainesville, FL). Bremsstrahlung photons were generated with a water-cooled copper plate positioned immediately after the 10 MeV electron beam scanning horn. A 0.003175 m (1/8 inch) polycarbonate sheet was placed above the samples during irradiation. Dose uniformity within the samples was confirmed by noting insignificant attenuation through the samples. Dosimetry was performed with GAFCHROMIC HD-810 dosimeters (International Specialty Products, Inc., Wayne, NJ) located immediately above (attached to the underside of the plastic sheet) and below (attached to the sample tray) the samples. Changes in optical density of the dosimetric film was measured with a spectrophotometer at 510 nm and correlated to absorbed dose with a previously established calibration curve. Accumulated dose was monitored throughout with the same dosimetric technique. Average accumulated dose per pass was approximately 0.0211 kGy.

Samples were separated into three groups, unirradiated (control), 1.0 and 3.0 kGy, respectively. Doses of 1.0 and 3.0 kGy were achieved with 47 and 142 passes, respectively. All samples were removed from the ice bath for the duration of radiation processing. After irradiation, all samples were returned to the ice bath and allowed to equilibrate to 0°C prior to thermal treatment.

During thermal treatment, each sample experienced four ambient conditions. Each condition was reflected in the heat transfer coefficients applied during simulation (Table 5-1). The four conditions were: (1) After removal from the ice bath and before application of steam in the retort (hAir), (2) during steam heating in the retort (hHeat), (3) during water cooling in the retort (hCool), and (4) placement in an ice bath (hIce), after removal from the retort. The method used to arrive at these values was described in Chapter 4.

Cans were exposed in duplicate to different retort processes. In general, three approximate set-point temperatures were used (104.4, 112.8, and 120.6°C) in conjunction with various heating times. Ambient temperatures in the retort were recorded with at least three thermocouples in proximity to the cans. Thermocouples were calibrated to a mercury-in-glass (MIG) thermometer installed on the retort. A vertical still-cook retort, supplied with 0.17 MPa (25 psig) steam in the pilot plant of the Department of Food Science and Human Nutrition at the University of Florida (Gainesville, FL) was used for this purpose.

All samples were kept in an ice bath after treatment, and were assayed for surviving spores within 48 hours. The contents of each can were placed into a sterile

250 ml beaker and mixed well for several minutes by hand. The puree was sampled in triplicate by transferring ca. 0.5 grams of puree into respective test tubes with sterile, disposable 10 ml pipettes. Samples were weighed on a digital scale, and immediately diluted by a factor of 10 (by weight) with sterile 0.15 M potassium phosphate buffer (pH 7.0). Serial dilutions were subsequently performed. Four 50 µl aliquots of respective dilutions were placed on petri dishes containing non-selective media. The drops were dried on the surface of the media in a 55°C incubator. The plates were arranged in groups of four, covered lightly with aluminum foil, inverted, and incubated at 55°C. Colonies were counted manually after 24 hours of incubation.

The recovery medium consisted of 8 grams of nutrient broth (Fisher Scientific, Pittsburgh, PA) and 20 grams of Difco Bacto Agar (Fisher Scientific, Pittsburgh, PA) per liter of distilled water. The medium was autoclave sterilized, and poured into sterile petri dishes (Fisher Scientific, Pittsburgh, PA). After cooling and hardening, the plates were stored in a refrigerator. All plates were used within two weeks of preparation.

The PEIE method was used to determine thermal inactivation Arrhenius kinetic parameters. The PEIE method was described in Chapter 4 for the case of dynamic thermal treatments with samples that heat nonuniformly. Briefly, the underlying process of the PEIE method is depicted in Fig. 5-1. The figure depicts two iterations of the method for two non-isothermal processes (indicated by "Process 1" and "Process 2") for which extents of reaction for a given reactant are known. For the first iteration, two $E_a$ values are arbitrarily selected ($E_a$-1a and $E_a$-1b). For each process the lower $E_a$

value, $E_a$-1b, yielded curves with shallower slopes, while the higher value, $E_a$-1a, yielded curves with steeper slopes. Points of intersection of respective curves, represent equivalent isothermal processes for these two arbitrarily selected $E_a$ values. An estimation of the *apparent* $E_a$ value is obtained by combining the equivalent isothermal process specifications to measured extent-of-reaction data. New $E_a$ value estimates, $E_a$-2a and $E_a$-2b, are chosen in the range of this $E_a$ value estimate, and the procedure repeated. Ultimately, the two $E_a$ values converge about the *apparent* $E_a$ value.

The explicit finite-difference method described in Chapter 3 was used to simulate heat transfer in the cans of puree. Extents of reaction in all volume elements were calculated using a temperature value that was 5% above the mean temperature for each simulated time interval. Can dimensions, thermophysical properties of the food, applicable heat transfer coefficients, and parameters related to numerical heat transfer simulation were provided in Table 5-1.

## Results and Discussion

Retort temperature profiles for thermal processes applied to samples irradiated to 1.0 and 3.0 kGy are shown in Fig. 5-2, Fig. 5-3 and Fig. 5-4. Resulting mass average spore survival ratios and pH of the puree are shown in Table 5-2. Complete microbial enumeration data were provided for 1.0 and 3.0 kGy in Appendix E and Appendix F, respectively. Corresponding data for 0 kGy were provided in Chapter 4 (see Appendix D).

Arrhenius thermal inactivation kinetic parameters were determined with the PEIE method. Data and results for the control case (0 kGy) were previously reported in Chapter 4. Results for preirradiation doses of 1.0 and 3.0 kGy are shown in Fig. 5-5, 5-6, 5-7, and 5-8. Mean inactivation rate constants at 121.1°C and activation energy values for pre-irradiation doses of 0, 1.0, and 3.0 kGy were 0.26, 0.28 and 0.38 $\text{min}^{-1}$ and 250, 200 and 200 kJ/mol, respectively. Frequency histograms of $\ln(k_r)$ and $E_a$ data for all doses showed normal distribution behavior. The origin of data points in Fig. 5-5 and Fig. 5-6 was provided in Appendix H. The origin of data points in Fig. 5-7 and Fig. 5-8 was provided in Appendix I.

In the vernacular of food scientists, the preceding Arrhenius kinetic parameters may be converted to $D_0$ (time required to achieve a one log cycle reduction in concentration at 121.1°C) and z (temperature change required to achieve a one log cycle change in $D_0$ value) values. Corresponding $D_0$ and z values for the temperature range 105 to 125°C for pre-irradiation treatments of 0, 1.0 and 3.0 kGy are 8.9, 8.2 and 6.0 minutes, and 11.4, 14.2 and 14.2 °C, respectively.

Fundamental differences between the PEIE method and traditional kinetic parameter estimation methods may preclude direct comparisons of estimated parameters. The following discussion will focus on the case of uniformly heating samples, since that is the typical premise of traditional isothermal kinetics experiments. Although microbial survivor curves often possess shoulders during initial phases of inactivation (Fig. 5-9), it is common practice to discard the shoulder, and obtain kinetic parameters from the straight portion of the curve. This is done by thermally activating spores prior to lethal exposure and/or by performing a linear regression on data only in the straight portion of the curve. As shown in Fig. 5-9, use of parameters derived from the straight portion of survivor curves will constantly over predict lethality to an unknown extent when shoulders are present during lethality.

As implemented, the PEIE method only considers irreversible first-order inactivation. Additionally, the PEIE method operates by connecting two points on a survivor curve (Fig. 5-9). Thus, in the absence of a shoulder on the actual survivor curve, the PEIE method would provide results similar to the traditional kinetic parameter estimation approach. When a shoulder does exist, the parameters obtained with the PEIE method will differ from those of the traditional approach. Reed et al. (1951) provided an excellent example of this difference. Amongst experiments involving several bacterial spores and suspending media, they performed experiments with pea puree inoculated with the same strain of spores used in this study. The authors reported the presence of shoulders in survivor curves, and reported results in the following two ways, (1) total time to achieve 99.99% reduction in spore count (four log-cycle reduction) at 121.1°C, and (2) time required for the straight line portion to traverse four log-cycles at 121.1°C. They reported 36 and 25.4 minutes, respectively, for the two methods; each with z values of 11.1°C. On a per log-cycle basis the reported times are equivalent to $D_0$ values, which are 9.0 and 6.3 minutes, respectively. Their first method of reporting results is analogous to the result provided by the PEIE method. On a $D_0$ and z value basis, the values reported here for unirradiated puree ($D_0$ = 8.9 min; z = 11.4°C) agreed very well with data from Reed et al. (1951) ($D_0$ = 9.0 min; z = 11.1°C).

The amount of thermal process reduction achieved for each irradiation dose may be calculated by specifying a process sterilizing value (SV) (the number of log cycle or decimal reductions in spore count that results from a given process). The heating time at 121.1°C required to achieve a specified SV is referred to as the $F_o$ value of the process. For an SV of 5, the process $F_o$ is calculated via (Teixeira, 1992):

$$F_o = SV \cdot D_0. \tag{5-2}$$

Thus, equivalently lethal $F_o$ values for 0, 1.0 and 3.0 kGy would be 44.5, 41 and 30 minutes. Fig. 5-10 shows simulated retort processes for cans of pea puree that achieve these $F_o$ values. The figure shows that for an equivalent level of lethality, retort steam may be shut off at 72, 65 and 54 minutes for cans pre-treated with 0, 1.0 and 3.0 kGy, respectively. Thus, for this particular product and target organism, 1.0 and 3.0 kGy provide 10 and 25% reductions, respectively, in thermal process cycle times. Such savings translate directly into enhanced final product quality, thermal process energy savings per unit of product and increased thermal processing capacity.

A comprehensive economic feasibility study of hybrid radiation-thermal processes is beyond the scope of this chapter. Selection of a particular process configuration requires consideration of factors involving the product and existing or expected production volumes. Two irradiation modes are commercially available, namely electron beam and Cobalt-60 gamma irradiation. The costs of installation (US$3MM to US$6MM) and operation of both types of facilities are comparable

(US$200,000 to US$400,000 per year), however, selection of an appropriate irradiation mode depends primarily on the geometry of the product and packaging materials used. Gamma radiation would be required for thick products and/or products in bulky metallic containers. Electron beam radiation would be suitable for thin products in polymer or fiber based packages.

The main benefit of electron beam irradiation is high power output, thus, high dose rate and high throughput. The main drawback of electron beam irradiation is a strict product thickness restriction. Maximum product thickness for a 10 MeV electron beam is in the range of 0.03 to 0.04 m (about 1.5 inches) for one sided exposure and about 0.09 m (about 3.5 inches) for two sided exposure. Using process efficiency assumptions, Diehl (1990) showed that in order to process 1.8 metric tons of food per hour with a dose of 10 kGy would require a 42 PBq (1.1 MCi) $Co^{60}$ facility or 10 kW of 10 MeV electrons. Increasing processing capacity requires increasing the amount of $Co^{60}$ or adding additional beam power by either adding machines or using a higher power machine. Typical $Co^{60}$ facilities are designed to use up to 115 to 191 PBq ( 3 to 5 MCi). Modern 10 MeV electron beam facilities operate in the range of 10 to 100 kW. Machines capable of 150 to 200 kW are expected to be commercially available in the near future.

Table 5-1. Data used for numerical heat transfer simulation of retort processed canned peas.

| | | | |
|---|---|---|---|
| Thermophysical properties: | | | |
| Thermal conductivity | k | 0.682 | W/(m·°C) |
| Density | $\rho$ | 958 | Kg/m$^3$ |
| Heat capacity | $c_p$ | 4,350 | J/(Kg·°C) |
| Heat Transfer Coefficients: | | | |
| Ambient in pilot plant | hAir | 1 | W/(m$^2$·°C) |
| Steam in retort | hHeat | 5,500 | W/(m$^2$·°C) |
| Cooling water in retort | hCool | 90 | W/(m$^2$·°C) |
| Ice bath | hIce | 60 | W/(m$^2$·°C) |
| Can Dimensions: | | | |
| Diameter | D | 0.0602 | m |
| Height | H | 0.0348 | m |
| Nodes: | | | |
| In radius | Nr | 12 | |
| In half-height | Ny | 12 | |

Table 5-2. Mass average survival ratios of *B. stearothermophilus* spores (C/Co) and overall pH values of canned pea puree after being subjected to retort processes described in Fig. 5-2, Fig. 5-3, and Fig. 5-4.

| Retort | Measured | | Survival Ratio (C /Co) | | | |
|---|---|---|---|---|---|---|
| Process | Final pH | | Measured | | Predicted | |
| | 1.0 kGy | 3.0 kGy | 1.0 kGy | 3.0 kGy | 1.0 kGy | 3.0 kGy |
| 1 | | | 0.609 | 0.539 | 0.701 | 0.618 |
| 2 | 6.12 | 6.19 | 0.578 | 0.530 | 0.494 | 0.386 |
| 3 | 6.08 | 6.10 | 0.291 | 0.230 | 0.269 | 0.168 |
| 4 | 5.95 | 5.94 | 0.151 | 0.064 | 0.129 | 0.063 |
| 5 | | | 0.136 | 0.057 | 0.066 | 0.025 |
| 6 | | | 0.012 | 0.003 | 0.029 | 0.008 |
| 7 | | | 0.003 | | 0.022 | |
| 8 | | | 0.0004 | | 0.008 | |
| 9 | | | | 0.571 | | 0.815 |
| 10 | 6.33 | 6.37 | | 0.526 | | 0.657 |
| 11 | | | 0.503 | 0.438 | 0.550 | 0.422 |
| 12 | 6.23 | 6.27 | 0.376 | 0.295 | 0.409 | 0.305 |
| 13 | 6.22 | 6.20 | 0.319 | 0.210 | 0.273 | 0.179 |
| 14 | 6.16 | 6.13 | 0.135 | 0.089 | 0.121 | 0.059 |
| 15 | 6.39 | 6.45 | | 0.453 | | 0.680 |
| 16 | 6.32 | 6.28 | 0.425 | 0.334 | 0.476 | 0.396 |
| 17 | 6.22 | 6.21 | 0.252 | 0.254 | 0.321 | 0.239 |
| 18 | | | 0.321 | 0.193 | 0.264 | 0.185 |
| 19 | 6.22 | 6.17 | | 0.122 | | 0.107 |
| 20 | 6.18 | 6.15 | 0.117 | 0.085 | 0.116 | 0.063 |
| 21 | | | 0.075 | 0.036 | 0.064 | 0.029 |
| 22 | 6.12 | 6.14 | 0.012 | 0.003 | 0.022 | 0.011 |

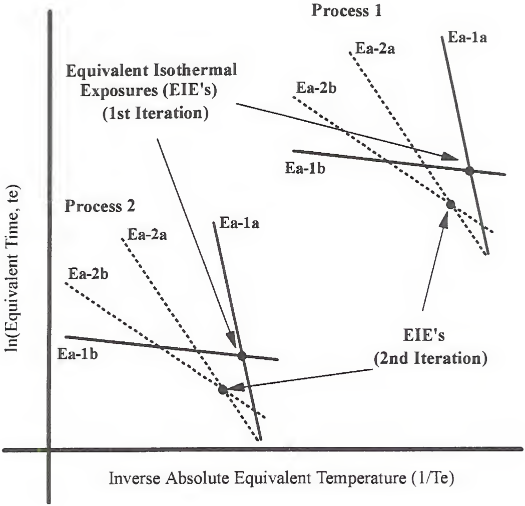


Figure 5-1. Graphical depiction of two iterations of the PEIE method.

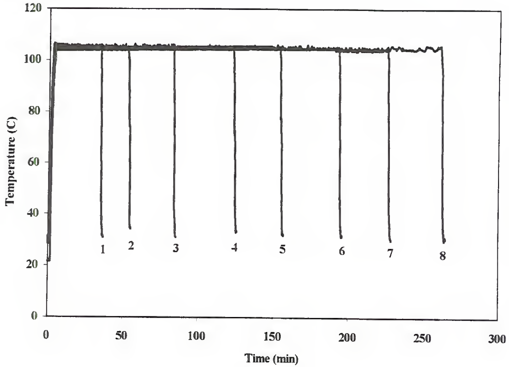


Figure 5-2. Transient thermal retort treatments applied to cans of inoculated pea puree.

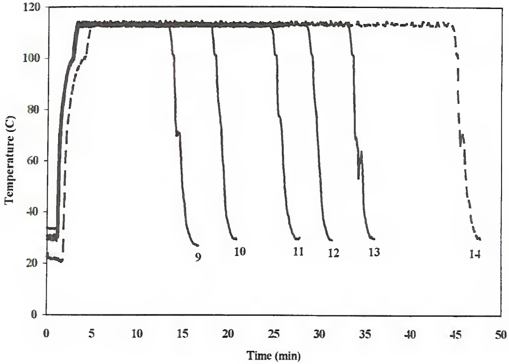


Figure 5-3. Additional transient thermal retort treatments applied to cans of inoculated pea puree.

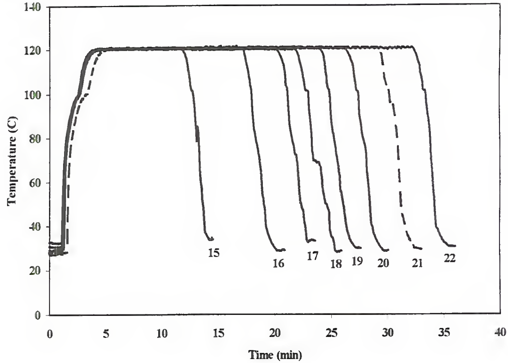


Figure 5-4. Additional transient thermal retort treatments applied to cans of inoculated pea puree.

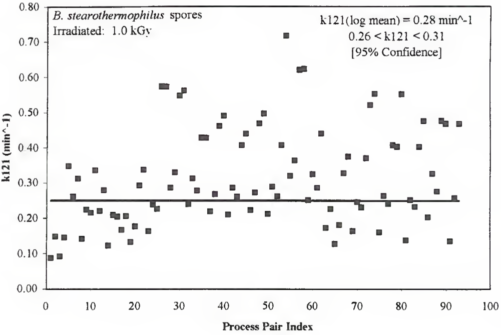


Figure 5-5. Values of thermal inactivation rate constant, k, at 121.1 °C for inactivation of spores of *B. stearothermophilus* in canned pea puree as determined with the PEIE method for samples pre-irradiated to 1.0 kGy.

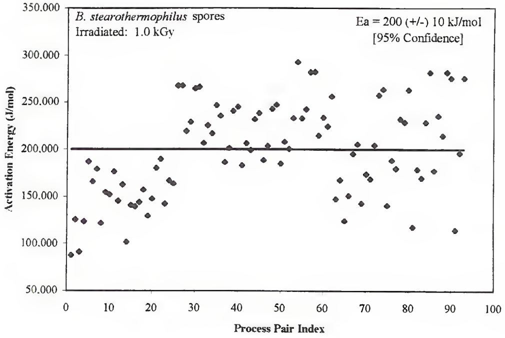


Figure 5-6. Activation energy, $E_a$, values for inactivation of spores of *B. stearothermophilus* in canned pea puree as determined with the PEIE method for samples irradiated to 1.0 kGy.

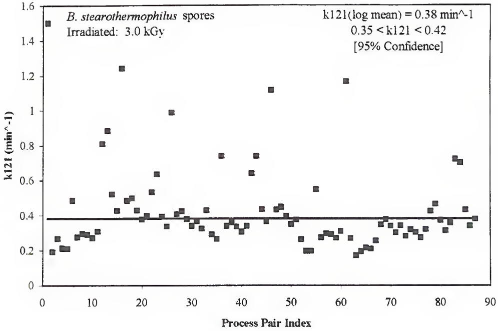


Figure 5-7 Values of thermal inactivation rate constant, k, at 121.1 °C for inactivation of spores of *B. stearothermophilus* in canned pea puree as determined with the PEIE method for samples pre-irradiated to 3.0 kGy.

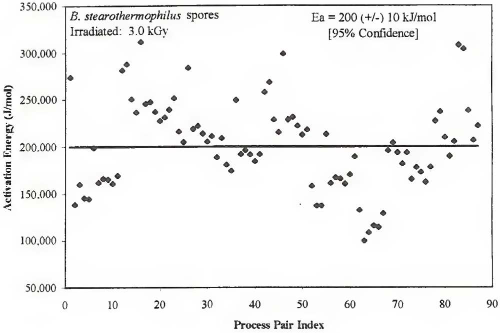


Figure 5-8. Activation energy, $E_a$, values for inactivation of spores of *B. stearothermophilus* in canned pea puree as determined with the PEIE method for samples irradiated to 3.0 kGy.

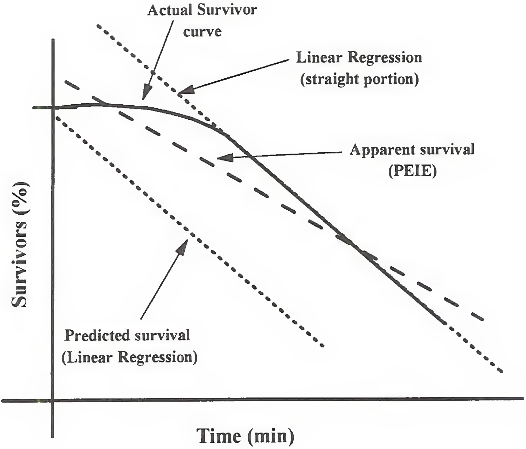

Actual Survivor curve
Linear Regression (straight portion)
Apparent survival (PEIE)
Predicted survival (Linear Regression)
Survivors (%)
Time (min)

Figure 5-9. Traditional isothermal kinetics typically uses the straight portion of survivor curves. The PEIE method assumes a straight line between two points on the actual survivor curve.

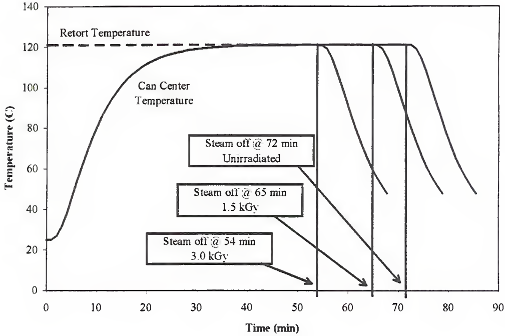


Figure 5-10. Retort processes delivering equivalent degrees of lethality for cans of pea puree pre-irradiated to 0, 1.0 and 3.0 kGy.

# CHAPTER 6
# SUMMARY AND CONCLUSIONS

The purpose of this work was to develop a method for estimating product and process specific thermal kinetic parameters in order to accurately define thermal processing requirements. A method, referred to as the Paired Equivalent Isothermal Exposures (PEIE) method was shown to determine kinetic parameters of reactants in foods that do and do not heat uniformly when subjected to dynamic thermal treatments.

An efficient numerical heat transfer algorithm was developed in order to apply the PEIE method to nonuniformly heating foods. Explicit finite difference algorithms are often criticized because of the time step restriction necessary to ensure a stable solution. It was shown that such criticism may be appropriate when the heat transfer problem in modeled in a manner that associates thermal mass with the two-dimensional surface of the object (referred to as the Capacitance Surface Node, CNS, approach), as is often the case in the literature. However, it was shown that by not associating thermal mass with the surface (referred to as the Non-Capacitance Surface Node, NCSN, approach), a more accurate physical description of the problem results, which also significantly alleviates the time step restriction.

Application of time discrete temperature data to thermal process calculations necessitates prudent time step selection. The NCSN approach is attractive, because it typically offers a larger time step than required for thermal process design tasks. It also offers ease of programming and fast execution associated with explicit finite difference algorithms.

Predicting the impact of a thermal process on thermally labile factors from numerical heat transfer methods requires appropriate use of discrete time-temperature data. Numerical heat transfer simulation algorithms generate temperature data at predetermined locations in the object and at specific points in time. The spatial locations at which temperatures are calculated are referred to as nodes. From each node's initial temperature, numerical heat transfer algorithms march forward in time by a specified time step. Thus, for any given node, temperature is known only before and after each time step. Thermally sensitive biochemical reactions occur during intervals of time between the instances when temperature is actually known. To predict extents of thermally sensitive reactions, it is often assumed that the temperature at the node is applicable to the entire volume element that contains the node. Extents of reactions are estimated for each time interval, such that the concentration of reactant at the end of one time interval serves as the initial concentration for the next interval. However, no guideline existed regarding which temperature should be used during each time interval. Although common practice probably involves the use of temperature values predicted for each node at the beginning or end of each time interval, it was shown that a temperature at or within 10% above the mean for each node and for each time interval is more appropriate. This guideline was developed for the case of constituents that follow first-order kinetics with Arrhenius temperature dependence. A NCSN based explicit finite difference algorithm employing the aforementioned guideline, was developed as a foundation for extending the concepts of the PEIE method to the case of foods that do not heat uniformly.

The PEIE method was successfully extended to apply to the case of nonuniformly heated products. The ability of the PEIE method to identify Arrhenius kinetic parameters from simulated retort experiments provided confidence that the method works as proposed. The PEIE method was used to obtain Arrhenius kinetic parameters for *Bacillus stearothermophilus* (FS 1518) spores in canned pea puree from actual mass average survival data. Since the PEIE method essentially assumes a straight-line between two points on an isothermal survivor curve, complex inactivation behavior, such as thermal activation of bacterial spores, which produce shoulders on isothermal survivor curves, was expected to influence results of the PEIE method. When such considerations were taken into account, it was shown that resulting parameters were similar to those derived from comparable data in the literature.

The PEIE method was employed in a thermal process design exercise in order to quantify reductions in thermal processing requirements due to irradiation pre-treatments. The product was canned pea puree inoculated with spores of *B. stearothermophilus*. Thermal kinetic parameters of the spores were determined for products irradiated with 0, 1.0 and 3.0 kGy prior to thermal treatment. Thermal process reductions, as indicated by simulated steam shut-off time, indicated savings of 10 and 25% over unirradiated product for products irradiated to 1.0 and 3.0 kGy, respectively.

The PEIE method offers a relatively simple means of defining product and process specific thermal kinetic parameters. The PEIE method may be particularly useful in the design of hybrid thermal processes, because it is able to quantify the impact of non-thermal pretreatments on thermal inactivation parameters on a product and process specific basis. Possibilities for further research include: (1) assessment of other non-isothermal pretreatments on microbial thermal sensitivity. Such treatments might include high pressure, ultra-sound, high voltage electric pulses, anti-bacterial agents, etc., (2) development of novel processing techniques, such as pH modifying compounds that deploy at critical moments during thermal processing, (3) comparing magnitudes of enhanced sensitivity for various microorganisms in different types of foods, (4) development of a simplified approach for applying the PEIE method to the cases of convection heating foods such as liquid foods with solid particulates and (5) advancing the concept the PEIE method to cases of more complex reaction behavior.

# APPENDIX A
# INVALID ASSUMPTION OF EPM

Let $G(E) = \int_{t_0}^{t_1} \exp\{-E / RT(t)\}\, dt$ and let $L_E$ be the line in the two variables $1/T$ and $\ln t$ defined by the equation $\ln t = \ln G(E) - E/RT$. Denote the set of all such lines by $\lambda$. If $E_0 \neq E_1$, the lines $L_{E0} = L_{E1}$ will have different slopes and will thus intersect at a unique point $L_{E0} \cap L_{E1}$. The intersection determines the time and temperature values for the equivalent isothermal exposure (EIE) to the thermal exposure defined by the temperature curve $T(t)$ using the reference activation energies $E_0$ and $E_1$. Swartzel (1982) claimed that all lines $L_E$ cross at the same point, i.e. that an EIE is independent of activation energy. However, assuming that $T$ is a continuous function, these EIE's will never be the same unless the process is itself isothermal. Thus, we have the following theorem:

**Theorem 1.** *Let $T$ be a continuous function on $[t_0, t_1]$. Then, if $T$ is not constant, no two pairs $\{L_0, L_1\}$ and $\{L_2, L_3\}$ of distinct lines in $\lambda$ will have a common intersection point unless $\{L_0, L_1\} = \{L_2, L_3\}$. Only when $T$ is constant will all lines in $\lambda$ will cross at the same point.*

*Proof.* Let $L_i = L_{Ei}$ for $0 \leq i \leq 3$. We begin by proving the first statement of the theorem for the special case where $E_0 = E_2$, $E_0 < E_1$, and $E_0 < E_3$.

$$f(E) = \frac{E - E_0}{R \ln G(E_0) / G(E)} \tag{A1}$$

for $E > E_0$. As shown in Eq. (9b), this is the value of $T$ for which the two lines $L_0$ and $L_E$ intersect. Note that if $f$ is a one-to-one function, then $L_0 \cap L_1 \neq L_0 \cap L_3$ unless $L_1 = L_3$ because the crossing points will have different $1/T$ coordinates.

**Proposition 2.** *Let $T$ be a continuous function on* $[t_0, t_1]$. *Then $f$ is a one-to-one function on the interval* $(E_0, \infty)$ *if and only if $T$ is not constant. Moreover, if $T$ is constant, so is $f$.*

*Proof.* We begin with a lemma.

**Lemma 3.** *The function* $(\ln G)'(E)$ *is strictly increasing if $T$ is not constant.*

*Proof.* We will show that

$$(\ln G)''(E) = \frac{G''(E)G(E) - G'(E)^2}{G(E)^2} > 0 \tag{A2}$$

for all $E \in \mathbf{R}$. Let $Z = Z(E, t) = \exp\{-E/RT(t)\}$, so that $G(E) = \int_{t_0}^{t} Z \, dt$. Since $Z$, $\partial Z/\partial E = -Z/RT$ and $\partial^2 Z/\partial E^2 = Z/R^2T^2$ are all continuous, we can differentiate under the integral sign to obtain $G'(E) = -\int_{t_0}^{t_1} Z / RT \, dt$ and $G''(E) = \int_{t_0}^{t_1} Z / R^2 T^2 \, dt$.

The Cauchy-Schwartz inequality applied to the continuous functions $h(t) = \exp\{-E/2RT\}/RT$ and $k(t) = \exp\{-E/2RT\}$ implies that

$$\begin{aligned} G'(E)^2 = \int_{t_0}^{t_1} \frac{Z}{RT} dt = \int_{t_0}^{t_1} h(t)k(t)\, dt &\le \left(\int_{t_0}^{t_1} h(t)^2 \, dt\right)\left(\int_{t_0}^{t_1} k(t)^2 \, dt\right) \\ &= \left(\int_{t_0}^{t_1} \frac{Z}{R^2T^2} dt\right)\left(\int_{t_0}^{t_1} Z \, dt\right) \\ &= G''(E)G(E) \end{aligned} \tag{A3}$$

with equality if and only if $h$ and $k$ are linearly dependent. We have $k(t) = RT(t)h(t)$, so $h$ and $k$ are proportional if and only if $T(t)$ is constant. Since $T$ is not constant by hypothesis, the inequality is strict. Thus, $G''(E)G(E) - G'(E)^2 > 0$ for any $E$, indicating that Eq. A2 and the lemma are satisfied.

Now, differentiating Eq. A1 gives

$$f'(E) = \frac{(\ln G(E_0) - \ln G(E)) + (E - E_0)(\ln G)'(E)}{R\,[\ln G(E_0) / G(E)]^2}. \tag{A4}$$

Suppose that $f$ is not one-to-one. Then there exists $E_1 > E_0$ such that $f'(E_1) = 0$, and we obtain

$$\frac{\ln G(E_1) - \ln G(E_0)}{E_1 - E_0} = (\ln G)'(E_1). \tag{A5}$$

By the mean value theorem, there also exists $E_2 \in (E_0, E_1)$ such that

$$\frac{\ln G(E_1) - \ln G(E_0)}{E_1 - E_0} = (\ln G)'(E_2). \tag{A6}$$

Since $E_1 \neq E_2$ this implies that $(\ln G)'$ is not on-to-one. By the lemma, $T$ is constant.

The converse is trivial; if $T = T_0$ is constant, it follows immediately that $G(E) = (t_1 - t_0)\exp\{-E/RT_0\}$ and $f(E) = T_0$.

**Corollary 4.** *If $T$ is a continuous, nonconstant function on $[t_0, t_1]$ and $E_0 < E_1$, $E_0 < E_3$ and $E_1 \neq E_3$, then $L_0 \cap L_1 \neq L_0 \cap L_3$.*

Now take two pairs $\{L_0, L_1\}$ and $\{L_2, L_3\}$ of distinct lines in $\lambda$, and suppose that $\{L_0, L_1\} \neq \{L_2, L_3\}$. Renumbering if necessary, we may assume $E_0 < E_1$, $E_2 < E_3$, and $E_0 \leq E_2$. If $L_0 \cap L_1 = L_2 \cap L_3$, then calling this common intersection point $P$, we have

$L_0 \cap L_1 \cap L_2 \cap L_3 = \{P\}$. But then $L_0 \cap L_1 = L_0 \cap L_3 = \{P\}$, implying $E_1 = E_3$ by the corollary. Since $\{L_0, L_1\} \neq \{L_2, L_3\}$, this gives $E_0 < E_2$. Applying the corollary again to the equality $L_0 \cap L_1 = L_0 \cap L_2$, we obtain $E_1 = E_2$. But this gives $E_2 = E_3$, a contradiction. Thus, $L_0 \cap L_1 \neq L_0 \cap L_3$, proving the first statement in the theorem.

Finally, it is obvious that if $T = T_0$ is constant, the lines in $\lambda$ have the common intersection point determined by $t = t_1 - t_0$ and $T = T_0$.

# APPENDIX B
# HEAT TRANSFER CODE--FINITE CYLINDER

```
'MODULE 1 of 4:  Heat transfer simulation for a finite cylinder.

'Module 1 performs input and output functions.  The program launches through
'subroutine MAIN, which calls subroutines CAPMAIN - for capacitance surface nodes,
'NOCAPMAIN for non-capacitance surface nodes and ANALMAIN for analytical
'solution.
'These subroutines launch modules 2, 3 and 4, which perform heat transfer simulations
'using capacitance surface nodes (finite difference), non-capacitance surface nodes
'(finite-difference), and an analytical solution.

Option Explicit

Global SolutionTime As Double
Global Nr As Integer, Ny As Integer
Global Alpha As Double, hConvection As Double, kT As Double
Global Density As Double, Cp As Double
Global InitialTemp As Double, AmbientTemp As Double
Global Radius As Double, Height As Double
Global T() As Double, r() As Double
Global TempAtrxt() As Double
Global StartAtRow As Integer
Global OutFile As String
Dim Start As Single, Finish As Single, Duration As Single

Sub Main ()

   Call SetInputVariables
      Alpha = kT / (Density * Cp)

Open OutFile For Append As #1
Print #1, "h="; ","; hConvection
Print #1, "time="; ","; SolutionTime; ","

   Call SetGrid
```

```
      Start = Timer
      Call CapMain
      Finish = Timer
      Duration = Finish - Start
         StartAtRow = 1
         Call Print2Grid
         Call Print2File


      Start = Timer
      Call NoCapMain
      Finish = Timer
      Duration = Finish - Start
         StartAtRow = 7
         Call Print2Grid
         Call Print2File


      Call AnalMain
         PrintAnal2File

   Screen.MousePointer = 1
Close #1

End Sub

Sub Print2File ()

Dim i As Integer
   Print #1, Format(Duration, "fixed")
   For i = 1 To Nr
      Print #1, Format(r(i) / Radius, "fixed"); ","; T(i, 1)
   Next i

End Sub

Sub Print2Grid ()

   Dim i As Integer

   frmOutPut.grdOutPut.Row = StartAtRow
   For i = 1 To Nr
      frmOutPut.grdOutPut.Col = i
      frmOutPut.grdOutPut.Text = Format(r(i) / Radius, "fixed")
    Next i

    frmOutPut.grdOutPut.Row = StartAtRow + 1
    For i = 1 To Nr
      frmOutPut.grdOutPut.Col = i
      frmOutPut.grdOutPut.Text = Format(T(i, 1), "Scientific")
    Next i

frmOutPut.Refresh
End Sub


Sub PrintAnal2File ()
Dim j As Double

For j = 1 To 11
   Print #1, Format(Radius, "fixed"); ","; TempAtrxt(j)
Next j

End Sub

Sub SetGrid ()

   Dim i As Integer

   frmOutPut.grdOutPut.Rows = 20
   frmOutPut.grdOutPut.Cols = Nr + 1

   frmOutPut.grdOutPut.FixedRows = 1
   frmOutPut.grdOutPut.FixedCols = 1

   For i = 0 To Nr
      Select Case Nr
         Case Is < 11
            frmOutPut.grdOutPut.ColWidth(i) = frmOutPut.grdOutPut.Width / (Nr +
1.2)
         Case Else
            frmOutPut.grdOutPut.ColWidth(i) = frmOutPut.grdOutPut.Width / (10 +
1.2)
      End Select
   Next i

   For i = 1 To Nr
```

```
      frmOutPut.grdOutPut.ColAlignment(i) = 1
      frmOutPut.grdOutPut.FixedAlignment(i) = 2
   Next i

   frmOutPut.grdOutPut.Row = 0
   For i = 1 To Nr
      frmOutPut.grdOutPut.Col = i
      frmOutPut.grdOutPut.Text = i
   Next i

   frmOutPut.grdOutPut.Col = 0
      For i = 2 To 14
         frmOutPut.grdOutPut.Row = i
         Select Case i
            Case 2
               frmOutPut.grdOutPut.Text = "Cap"
            Case 8
               frmOutPut.grdOutPut.Text = "NoCap"
            Case 13
               frmOutPut.grdOutPut.Text = "Anal"
         End Select
      Next i

   frmOutPut.Refresh

End Sub

Sub SetInputVariables ()

   InitialTemp = Val(frmInPut.txtInitialT.Text)
   AmbientTemp = Val(frmInPut.txtAmbientT.Text)

   Nr = Val(frmInPut.txtRnodes.Text)
   Ny = Val(frmInPut.txtHnodes.Text)

   Radius = Val(frmInPut.txtRadius.Text) * 2.54
   Height = Val(frmInPut.txtHeight.Text) * 2.54 / 2

   kT = Val(frmInPut.txtK.Text)
   Cp = Val(frmInPut.txtCp.Text)
   Density = Val(frmInPut.txtDensity.Text)
   hConvection = Val(frmInPut.txth.Text)

   SolutionTime = Val(frmInPut.txtSolutionTime.Text) * 60
```

```
    OutFile = frmInPut.txtOutFile

End Sub
```

MODULE 2 of 4: Capacitance surface nodes.

```
Option Explicit

Dim i As Integer, j As Integer
Dim y() As Double
Dim dr() As Double, rA() As Double
Dim dy() As Double, yA() As Double
Dim dVol() As Double, TimeStep As Double
Dim tPosition As Double, Told() As Double

Sub CapMain ()

  Call SetNodalDimensions

  TimeStep = MinAllowedTimeStep()

  Call SetInitialTemp

  Call SimulationLoop

End Sub

Private Function MinAllowedTimeStep () As Double

Dim i As Integer, j As Integer
Dim MinTime As Double, dtSmallest As Double
Dim UA As Double, UAr As Double, UAy As Double
Dim a1 As Double, a2 As Double, a3 As Double, a4 As Double
ReDim dt(9) As Double

'  bot left (can center)
   i = 1: j = 1
   a1 = rA(i, j) / dr(i)
   a2 = yA(i) / dy(j)
   dt(1) = dVol(i, j) / (Alpha * (a1 + a2))

'  bot right
   i = Nr: j = 1
   a1 = rA(i - 1, j) / dr(i - 1)
   a2 = yA(i) / dy(1)
   UA = hConvection * rA(i, j) / kT
   dt(2) = dVol(i, j) / (Alpha * (a1 + a2 + UA))
```

```
' top left (Can top center)
  i = 1: j = Ny
  a1 = rA(i, j) / dr(i)
  a2 = yA(i) / dy(j - 1)
  UA = hConvection * yA(i) / kT
  dt(3) = dVol(i, j) / (Alpha * (a1 + a2 + UA))

' top right
  i = Nr: j = Ny
  a1 = rA(i - 1, j) / dr(i - 1)
  a2 = yA(i) / dy(j - 1)
  UAy = hConvection * yA(i) / kT
  UAr = hConvection * rA(i, j) / kT
  dt(4) = dVol(i, j) / (Alpha * (a1 + a2 + UAy + UAr))

' bot interior
  j = 1
  a1 = rA(1, j) / dr(1)
  a2 = rA(2, j) / dr(2)
  a3 = yA(2) / dy(j)
  dt(5) = dVol(2, j) / (Alpha * (a1 + a2 + a3))
  For i = 3 To Nr - 1
     a1 = rA(i - 1, j) / dr(i - 1)
     a2 = rA(i, j) / dr(i)
     a3 = yA(i) / dy(j)
     MinTime = dVol(i, j) / (Alpha * (a1 + a2 + a3))
     If MinTime < dt(5) Then dt(5) = MinTime
  Next i

' vertical center line - interior
  i = 1: j = 2
  a1 = rA(i, j) / dr(i)
  a2 = yA(i) / dy(j)
  a3 = yA(i) / dy(j - 1)
  dt(6) = dVol(i, j) / (Alpha * (a1 + a2 + a3))

' Side surface - interior
  i = Nr: j = 2
  a1 = rA(i - 1, 2) / dr(i - 1)
  a2 = yA(i) / dy(j)
  a3 = yA(i) / dy(j - 1)
  UA = hConvection * rA(i, j) / kT
  dt(7) = dVol(i, j) / (Alpha * (a1 + a2 + a3 + UA))
```

```
 '  Top surface - interior
    a1 = rA(1, Ny) / dr(1)
    a2 = rA(2, Ny) / dr(2)
    a3 = yA(2) / dy(Ny - 1)
    UA = hConvection * yA(2) / kT
    dt(8) = dVol(2, Ny) / (Alpha * (a1 + a2 + UA))
    For i = 3 To Nr - 1
       a1 = rA(i - 1, Ny) / dr(i - 1)
       a2 = rA(i, Ny) / dr(i)
       a3 = yA(i) / dy(Ny - 1)
       UA = hConvection * yA(i) / kT
       MinTime = dVol(i, Ny) / (Alpha * (a1 + a2 + a3 + UA))
       If MinTime < dt(8) Then dt(8) = MinTime
    Next i

 '  Interior nodes
    j = 3
    a1 = rA(1, j) / dr(1)
    a2 = rA(2, j) / dr(2)
    a3 = yA(2) / dy(j)
    a4 = yA(2) / dy(j - 1)
    dt(9) = dVol(2, 2) / (Alpha * (a1 + a2 + a3 + a4))
    For i = 3 To Nr - 1
       a1 = rA(i - 1, j) / dr(i - 1)
       a2 = rA(i, j) / dr(i)
       a3 = yA(i) / dy(j)
       a4 = yA(i) / dy(j - 1)
       MinTime = dVol(i, 2) / (Alpha * (a1 + a2 + a3 + a4))
       If MinTime < dt(9) Then dt(9) = MinTime
    Next i

    dtSmallest = dt(1)
    For i = 2 To 9
       If dt(i) < dtSmallest Then dtSmallest = dt(i)
    Next i

    MinAllowedTimeStep = dtSmallest


End Function

Private Sub SetInitialTemp ()

   ReDim T(Nr, Ny), Told(Nr, Ny)
```

```
'  initialize temperature

   For j = 1 To Ny
      For i = 1 To Nr
         Told(i, j) = InitialTemp
      Next i
   Next j

End Sub

Sub SetNodalDimensions ()

Dim Pi As Double
ReDim rA(Nr, Ny), yA(Nr), dVol(Nr, Ny)
ReDim r(Nr), y(Ny)
ReDim dr(Nr - 1), dy(Ny - 1)
ReDim rALocation(Nr), yALocation(Ny)
Dim yAsum As Double
Dim Check As Double

Pi = 4 * Atn(1)

'  set internodal distances in radial direction
   dr(1) = Radius / (Nr - 1)
   For i = 2 To Nr - 1
      dr(i) = dr(1)
   Next i

'  set internodal dist's in height direction
   dy(1) = Height / (Ny - 1)
   For i = 2 To Ny - 1
      dy(i) = dy(1)
   Next i

'  set radial node locations
   r(1) = 0
   For i = 2 To Nr
      r(i) = r(i - 1) + dr(i - 1)
   Next i

'  set locations for sides of vol elements - (log mean radius)
   rALocation(1) = r(2) / 2
   For i = 2 To Nr - 1
```

```
'       rALocation(i) = (r(i + 1) + r(i)) / 2
        rALocation(i) = (r(i + 1) - r(i)) / (Log(r(i + 1) / r(i)))
    Next i
    rALocation(Nr) = Radius

'   set height node locations
    y(1) = 0
    For i = 2 To Ny
        y(i) = y(i - 1) + dy(i - 1)
    Next i

'   set locations for tops and bottoms of elements
    yALocation(0) = 0
    For j = 1 To Ny - 1
        yALocation(j) = (y(j + 1) + y(j)) / 2
    Next j
    yALocation(Ny) = Height

'   Vertical areas - Radial heat flow
'    Check = 0
    For j = 1 To Ny
        For i = 1 To Nr
            rA(i, j) = 2 * Pi * rALocation(i) * (yALocation(j) - yALocation(j - 1))
'       If i = Nr Then Check = Check + rA(i, j)
        Next i
    Next j


'   Horizontal Areas - tops of elements - vertical heat flow
    yA(1) = Pi * rALocation(1) * rALocation(1)
    yAsum = yA(1)
    For i = 2 To Nr
        yA(i) = Pi * rALocation(i) * rALocation(i) - yAsum
        yAsum = yAsum + yA(i)
    Next i

'   Volumes
    Check = 0
    For j = 1 To Ny
        For i = 1 To Nr
                dVol(i, j) = yA(i) * (yALocation(j) - yALocation(j - 1))
                Check = Check + dVol(i, j)
        Next i
    Next j
```

```
End Sub

Private Sub SimulationLoop ()

Dim TermCoef As Double
Dim a1 As Double, a2 As Double, a3 As Double, a4 As Double
Dim UAr As Double, UAy As Double

tPosition = 0

Do
'  Check if time to display results
   Select Case SolutionTime - tPosition
      Case Is < TimeStep
'        Adjust last time step to land on desired time
         TimeStep = SolutionTime - tPosition
      Case 0
'        Display results now
         Exit Sub
   End Select

tPosition = tPosition + TimeStep

For j = Ny To 1 Step -1
   For i = Nr To 1 Step -1
   Select Case j
      Case 1
         TermCoef = Alpha * TimeStep / dVol(i, j)
         Select Case i
            Case 1
               a1 = rA(i, j) / dr(i)
               a2 = yA(i) / dy(j)
               T(i, j) = Told(i, j) * (1 - TermCoef * (a1 + a2)) + TermCoef * (a1 *
Told(i + 1, j) + a2 * Told(i, j + 1))
            Case 2 To Nr - 1
               a1 = rA(i - 1, j) / dr(i - 1)
               a2 = rA(i, j) / dr(i)
               a3 = yA(i) / dy(j)
               T(i, j) = Told(i, j) * (1 - TermCoef * (a1 + a2 + a3)) + TermCoef * (a1 *
Told(i - 1, j) + a2 * Told(i + 1, j) + a3 * Told(i, j + 1))
            Case Nr
               a1 = rA(i - 1, j) / dr(i - 1)
               a2 = yA(i) / dy(j)
```

```
          UAr = hConvection * rA(i, j) / kT
          T(i, j) = Told(i, j) * (1 - TermCoef * (a1 + a2 + UAr)) + TermCoef * (a1
* Told(i - 1, j) + a2 * Told(i, j + 1) + AmbientTemp * UAr)
      End Select
    Case 2 To Ny - 1
      TermCoef = Alpha * TimeStep / dVol(i, j)
      Select Case i
        Case 1
          a1 = rA(i, j) / dr(i)
          a2 = yA(i) / dy(j - 1)
          a3 = yA(i) / dy(j)
          T(i, j) = Told(i, j) * (1 - TermCoef * (a1 + a2 + a3)) + TermCoef * (a1 *
Told(i + 1, j) + a2 * Told(i, j - 1) + a3 * Told(i, j + 1))
        Case 2 To Nr - 1
          a1 = rA(i - 1, j) / dr(i - 1)
          a2 = rA(i, j) / dr(i)
          a3 = yA(i) / dy(j - 1)
          a4 = yA(i) / dy(j)
          T(i, j) = Told(i, j) * (1 - TermCoef * (a1 + a2 + a3 + a4)) + TermCoef *
(a1 * Told(i - 1, j) + a2 * Told(i + 1, j) + a3 * Told(i, j - 1) + a4 * Told(i, j + 1))
        Case Nr
          a1 = rA(i - 1, j) / dr(i - 1)
          a2 = yA(i) / dy(j - 1)
          a3 = yA(i) / dy(j)
          UAr = hConvection * rA(i, j) / kT
          T(i, j) = Told(i, j) * (1 - TermCoef * (a1 + a2 + a3 + UAr)) + TermCoef *
(a1 * Told(i - 1, j) + a2 * Told(i, j - 1) + a3 * Told(i, j + 1) + AmbientTemp * UAr)
      End Select
    Case Ny
      TermCoef = Alpha * TimeStep / dVol(i, j)
      Select Case i
        Case 1
          a1 = rA(i, j) / dr(i)
          a2 = yA(i) / dy(j - 1)
          UAy = hConvection * yA(i) / kT
          T(i, j) = Told(i, j) * (1 - TermCoef * (a1 + a2 + UAy)) + TermCoef * (a1
* Told(i + 1, j) + a2 * Told(i, j - 1) + AmbientTemp * UAy)
        Case 2 To Nr - 1
          a1 = rA(i - 1, j) / dr(i - 1)
          a2 = rA(i, j) / dr(i)
          a3 = yA(i) / dy(j - 1)
          UAy = hConvection * yA(i) / kT
          T(i, j) = Told(i, j) * (1 - TermCoef * (a1 + a2 + a3 + UAy)) + TermCoef *
(a1 * Told(i - 1, j) + a2 * Told(i + 1, j) + a3 * Told(i, j - 1) + AmbientTemp * UAy)
```

```
            Case Nr
               a1 = rA(i - 1, j) / dr(i - 1)
               a2 = yA(i) / dy(j - 1)
               UAy = hConvection * yA(i) / kT
               UAr = hConvection * rA(i, j) / kT
               T(i, j) = Told(i, j) * (1 - TermCoef * (a1 + a2 + UAy + UAr)) +
TermCoef * (a1 * Told(i - 1, j) + a2 * Told(i, j - 1) + AmbientTemp * UAy +
AmbientTemp * UAr)
         End Select
      End Select
   Next i
Next j

'  Update Temp grid
For j = 1 To Ny
   For i = 1 To Nr
      Told(i, j) = T(i, j)
   Next i
Next j

frmOutPut.txtTime.Text = Format(tPosition / 60, "fixed")

Loop

End Sub
```

MODULE 3 of 4: Non-capacitance surface nodes

```
Option Explicit

Dim i As Integer, j As Integer
Dim y() As Double
Dim dr() As Double, rA() As Double
Dim dy() As Double, yA() As Double
Dim dVol() As Double
Dim TimeStep As Double
Dim tPosition As Double, Told() As Double

Private Function MinAllowedTimeStep () As Double

Dim i As Integer, j As Integer
Dim MinTime As Double, dtSmallest As Double
Dim UA As Double, UAr As Double, UAy As Double
Dim a1 As Double, a2 As Double, a3 As Double, a4 As Double
ReDim dt(9) As Double

'  bot left (can center)
   i = 1: j = 1
   a1 = rA(i, j) / dr(i)
   a2 = yA(i) / dy(j)
   dt(1) = dVol(i, j) / (Alpha * (a1 + a2))

'  bot right
   i = Nr - 1: j = 1
   a1 = rA(i - 1, j) / dr(i - 1)
   a2 = yA(i) / dy(j)
   UA = 1 / (kT / (hConvection * rA(i + 1, j)) + dr(i) / rA(i, j))
   dt(2) = dVol(i, j) / (Alpha * (a1 + a2 + UA))

'  top left (Can top center)
   i = 1: j = Ny - 1
   a1 = rA(i, j + 1) / dr(i)
   a2 = yA(i) / dy(j - 1)
   UA = 1 / (kT / (hConvection * yA(i)) + dy(j) / yA(i))
   dt(3) = dVol(i, j) / (Alpha * (a1 + a2 + UA))

'  top right
   i = Nr - 1: j = Ny - 1
   a1 = rA(i - 1, j + 1) / dr(i - 1)
   a2 = yA(i) / dy(j - 1)
```

```
    UAy = 1 / (kT / (hConvection * yA(i)) + dy(j) / yA(i))
    UAr = 1 / (kT / (hConvection * rA(i + 1, j)) + dr(i) / rA(i, j))
    dt(4) = dVol(i, j) / (Alpha * (a1 + a2 + UAy + UAr))

'   bot interior
    j = 1
    a1 = rA(1, j) / dr(1)
    a2 = rA(2, j) / dr(2)
    a3 = yA(2) / dy(j)
    dt(5) = dVol(2, j) / (Alpha * (a1 + a2 + a3))
    For i = 3 To Nr - 2
       a1 = rA(i - 1, j) / dr(i - 1)
       a2 = rA(i, j) / dr(i)
       a3 = yA(i) / dy(j)
       MinTime = dVol(i, j) / (Alpha * (a1 + a2 + a3))
       If MinTime < dt(5) Then dt(5) = MinTime
    Next i

'   vertical center line - interior
    i = 1: j = 2
    a1 = rA(i, j) / dr(i)
    a2 = yA(i) / dy(j)
    dt(6) = dVol(i, j) / (Alpha * (a1 + 2 * a2))

'   Side surface - interior
    i = Nr - 1: j = 2
    a1 = rA(i - 1, j) / dr(i - 1)
    a2 = yA(i) / dy(j)
    UA = 1 / (kT / (hConvection * rA(i + 1, j)) + dr(i) / rA(i, j))
    dt(7) = dVol(i, j) / (Alpha * (a1 + 2 * a2 + UA))

'   Top surface - interior
    j = Ny - 1
    a1 = rA(1, j) / dr(1)
    a2 = rA(2, j) / dr(2)
    a3 = yA(2) / dy(j - 1)
    UA = 1 / (kT / (hConvection * yA(2)) + dy(j) / yA(2))
    dt(8) = dVol(2, j) / (Alpha * (a1 + a2 + a3 + UA))
    For i = 3 To Nr - 2
       a1 = rA(i - 1, j) / dr(i - 1)
       a2 = rA(i, j) / dr(i)
       a3 = yA(i) / dy(j - 1)
       UA = 1 / (kT / (hConvection * yA(i)) + dy(j) / yA(i))
       MinTime = dVol(i, j) / (Alpha * (a1 + a2 + a3 + UA))
```

```
        If MinTime < dt(8) Then dt(8) = MinTime
    Next i

'   Interior nodes
    j = 3
    a1 = rA(1, j) / dr(1)
    a2 = rA(2, j) / dr(2)
    a3 = yA(1) / dy(j)
    a4 = yA(1) / dy(j - 1)
    dt(9) = dVol(2, j) / (Alpha * (a1 + a2 + a3 + a4))
    For i = 3 To Nr - 2
        a1 = rA(i - 1, j) / dr(i - 1)
        a2 = rA(i, j) / dr(i)
        a3 = yA(i) / dy(j)
        a4 = yA(i) / dy(j - 1)
        MinTime = dVol(i, j) / (Alpha * (a1 + a2 + a3 + a4))
        If MinTime < dt(9) Then dt(9) = MinTime
    Next i

    dtSmallest = dt(1)
    For i = 2 To 9
        If dt(i) < dtSmallest Then dtSmallest = dt(i)
    Next i

    MinAllowedTimeStep = dtSmallest

End Function

Sub NoCapMain ()

    Call SetNodalDimensions

    TimeStep = MinAllowedTimeStep()

    Call SetInitialTemp

    Call SimulationLoop

End Sub

Private Sub SetInitialTemp ()

ReDim T(Nr, Ny), Told(Nr, Ny)
```

```
' initialize temperatures

  For j = 1 To Ny - 1
    For i = 1 To Nr - 1
      Told(i, j) = InitialTemp
    Next i
  Next j

'  For i = 1 To Nr - 1
'    Told(i, Ny) = InitialTemp
'  Next i

End Sub

Private Sub SetNodalDimensions ()

ReDim rA(Nr, Ny), yA(Nr - 1), dVol(Nr - 1, Ny - 1)
ReDim r(Nr), y(Ny)
ReDim dr(Nr - 1), dy(Ny - 1)
ReDim rALocation(Nr) As Double, yALocation(Ny) As Double

Dim yAsum As Double
Dim Pi As Double
Dim CheckVol

  Pi = 4 * Atn(1)

' set internodal distances in radial direction
  dr(Nr - 1) = Radius / (2 * Nr - 3):  ' Smallest Common 'dr' Unit
  For i = 1 To Nr - 2
    dr(i) = 2 * dr(Nr - 1)
  Next i

' set internodal dist's in height direction
  dy(Ny - 1) = Height / (2 * Ny - 3): ' Smallest common 'dy' Unit
  For i = 1 To Ny - 2
    dy(i) = 2 * dy(Ny - 1)
  Next i

' set radial node locations
  r(1) = 0
  For i = 2 To Nr
    r(i) = r(i - 1) + dr(i - 1)
  Next i
```

```
' set volume element bdy locations-(log mean radius)
  rALocation(1) = r(2) / 2
  For i = 2 To Nr - 1
'     rALocation(i) = (r(i + 1) + r(i)) / 2
     rALocation(i) = (r(i + 1) - r(i)) / (Log(r(i + 1) / r(i)))
  Next i
  rALocation(Nr) = Radius

' set height node locations
  y(1) = 0
  For i = 2 To Ny
     y(i) = y(i - 1) + dy(i - 1)
  Next i

' set locations for tops and bottoms of elements
  yALocation(1) = y(2) / 2
  For j = 2 To Ny - 1
     yALocation(j) = yALocation(j - 1) + dy(j - 1)
  Next j
  yALocation(Ny) = Height


' Vertical areas - Radial heat flow rA(Nr, top/bottom edge=0 or interior=1)
  For j = 1 To Ny
     For i = 1 To Nr
        rA(i, j) = 2 * Pi * rALocation(i) * (yALocation(j) - yALocation(j - 1))
     Next i
  Next j


' Horizontal Areas - tops of elements - vertical heat flow
  yA(1) = Pi * rALocation(1) * rALocation(1)
  yAsum = yA(1)
  For i = 2 To Nr - 2
     yA(i) = Pi * rALocation(i) * rALocation(i) - yAsum
     yAsum = yAsum + yA(i)
  Next i
  yA(Nr - 1) = Pi * Radius * Radius - yAsum

' Volumes
  CheckVol = 0
  For j = 1 To Ny - 1
     For i = 1 To Nr - 1
```

```
            dVol(i, j) = yA(i) * (yALocation(j) - yALocation(j - 1))
            CheckVol = CheckVol + dVol(i, j)
      Next i
   Next j

End Sub

Private Sub SimulationLoop ()

Dim TermCoef As Double
Dim a1 As Double, a2 As Double, a3 As Double, a4 As Double
Dim UAr As Double, UAy As Double

tPosition = 0

Do
'  Check if time to display results
   Select Case SolutionTime - tPosition
      Case Is < TimeStep
'        Adjust last time step to land on desired time
         TimeStep = SolutionTime - tPosition
      Case 0
'        Display results now
         Exit Sub
   End Select

tPosition = tPosition + TimeStep

For j = 1 To Ny - 1
   For i = 1 To Nr - 1
   Select Case j
      Case 1
         TermCoef = Alpha * TimeStep / dVol(i, j)
         Select Case i
            Case 1
               a1 = rA(i, j) / dr(i)
               a2 = yA(i) / dy(j)
               T(i, j) = Told(i, j) * (1 - TermCoef * (a1 + a2)) + TermCoef * (a1 *
Told(i + 1, j) + a2 * Told(i, j + 1))
            Case 2 To Nr - 2
               a1 = rA(i - 1, j) / dr(i - 1)
               a2 = rA(i, j) / dr(i)
               a3 = yA(i) / dy(j)
```

```
          T(i, j) = Told(i, j) * (1 - TermCoef * (a1 + a2 + a3)) + TermCoef * (a1 *
Told(i - 1, 1) + a2 * Told(i + 1, 1) + a3 * Told(i, j + 1))
        Case Nr - 1
          a1 = rA(i - 1, j) / dr(i - 1)
          a2 = yA(i) / dy(j)
          UAr = 1 / (kT / (hConvection * rA(i + 1, j)) + dr(i) / rA(i, j))
          T(i, j) = Told(i, j) * (1 - TermCoef * (a1 + a2 + UAr)) + TermCoef * (a1
* Told(i - 1, j) + a2 * Told(i, j + 1) + AmbientTemp * UAr)
          T(i + 1, j) = (kT * rA(i, j) * T(i, j) / dr(i) + hConvection * rA(i + 1, j) *
AmbientTemp) / (hConvection * rA(i + 1, j) + kT * rA(i, j) / dr(i))
      End Select
    Case 2 To Ny - 2
      TermCoef = Alpha * TimeStep / dVol(i, j)
      Select Case i
        Case 1
          a1 = rA(i, j) / dr(i)
          a2 = yA(i) / dy(j - 1)
          a3 = yA(i) / dy(j)
          T(i, j) = Told(i, j) * (1 - TermCoef * (a1 + a2 + a3)) + TermCoef * (a1 *
Told(i + 1, j) + a2 * Told(i, j - 1) + a3 * Told(i, j + 1))
        Case 2 To Nr - 2
          a1 = rA(i - 1, j) / dr(i)
          a2 = rA(i, j) / dr(i)
          a3 = yA(i) / dy(j - 1)
          a4 = yA(i) / dy(j)
          T(i, j) = Told(i, j) * (1 - TermCoef * (a1 + a2 + a3 + a4)) + TermCoef *
(a1 * Told(i - 1, j) + a2 * Told(i + 1, j) + a3 * Told(i, j - 1) + a4 * Told(i, j + 1))
        Case Nr - 1
          a1 = rA(i - 1, j) / dr(i - 1)
          a2 = yA(i) / dy(j - 1)
          a3 = yA(i) / dy(j)
          UAr = 1 / (kT / (hConvection * rA(i + 1, j)) + dr(i) / rA(i, j))
          T(i, j) = Told(i, j) * (1 - TermCoef * (a1 + a2 + a3 + UAr)) + TermCoef *
(a1 * Told(i - 1, j) + a2 * Told(i, j - 1) + a3 * Told(i, j - 1) + AmbientTemp * UAr)
          T(i + 1, j) = (kT * rA(i, j) * T(i, j) / dr(i) + hConvection * rA(i + 1, j) *
AmbientTemp) / (hConvection * rA(i + 1, j) + kT * rA(i, j) / dr(i))
      End Select
    Case Ny - 1
      TermCoef = Alpha * TimeStep / dVol(i, j)
      Select Case i
        Case 1
          a1 = rA(i, j) / dr(i)
          a2 = yA(i) / dy(j - 1)
          UAy = 1 / (kT / (hConvection * yA(i)) + dy(j) / yA(i))
          T(i, j) = Told(i, j) * (1 - TermCoef * (a1 + a2 + UAy)) + TermCoef * (a1
* Told(i + 1, j) + a2 * Told(i, j - 1) + AmbientTemp * UAy)
          T(i, j + 1) = (kT * T(i, j) / dy(j) + hConvection * AmbientTemp) /
(hConvection + kT / dy(j))
        Case 2 To Nr - 2
          a1 = rA(i - 1, j) / dr(i - 1)
          a2 = rA(i, j) / dr(i)
          a3 = yA(i) / dy(j - 1)
          UAy = 1 / (kT / (hConvection * yA(i)) + dy(j) / yA(i))
          T(i, j) = Told(i, j) * (1 - TermCoef * (a1 + a2 + a3 + UAy)) + TermCoef *
(a1 * Told(i - 1, j) + a2 * Told(i + 1, j) + a3 * Told(i, j - 1) + AmbientTemp * UAy)
          T(i, j + 1) = (kT * T(i, j) / dy(j) + hConvection * AmbientTemp) /
(hConvection + kT / dy(j))
        Case Nr - 1
          a1 = rA(i - 1, j) / dr(i - 1)
          a2 = yA(i) / dy(j - 1)
          UAy = 1 / (kT / (hConvection * yA(i)) + dy(j) / yA(i))
          UAr = 1 / (kT / (hConvection * rA(i + 1, j)) + dr(i) / rA(i, j))
          T(i, j) = Told(i, j) * (1 - TermCoef * (a1 + a2 + UAy + UAr)) +
TermCoef * (a1 * Told(i - 1, j) + a2 * Told(i, j - 1) + AmbientTemp * (UAy + UAr))
          T(i, j + 1) = (kT * T(i, j) / dy(j) + hConvection * AmbientTemp) /
(hConvection + kT / dy(j))
          T(i + 1, j) = (kT * rA(i, j) * T(i, j) / dr(i) + hConvection * rA(i + 1, j) *
AmbientTemp) / (hConvection * rA(i + 1, j) + kT * rA(i, j) / dr(i))
          T(Nr, Ny) = (T(i, j + 1) + T(i + 1, j)) / 2
      End Select
    End Select
  Next i
Next j

' Update Temp Grid
For j = 1 To Ny
  For i = 1 To Nr
    Told(i, j) = T(i, j)
  Next i
Next j

frmOutPut.txtTime.Text = Format(tPosition / 60, "fixed")

Loop

End Sub
```

MODULE 4 of 4: Analytical solution

```
Option Explicit
Dim eps As Double
Dim BiotCyl As Double, BiotSlab As Double
Dim rPosition As Double, tPosition As Double, xPosition As Double
Dim i As Integer, j As Integer, Pi As Double
Dim thetaCyl As Double, thetaSlab As Double
Dim TransRoot() As Double
Dim NumRoots As Integer

Const BJ0p1 = 1
Const BJ0p2 = -.001098628627
Const BJ0p3 = .00002734510407
Const BJ0p4 = -.000002073370639
Const BJ0p5 = 2.093887211E-07
Const BJ0q1 = -.01562499995
Const BJ0q2 = .0001430488765
Const BJ0q3 = -.000006911147651
Const BJ0q4 = .000007621095161
Const BJ0q5 = -9.34945152E-08
Const BJ0r1 = 57568490574#
Const BJ0r2 = -13362590354#
Const BJ0r3 = 651619640.7
Const BJ0r4 = -11214424.18
Const BJ0r5 = 77392.33017
Const BJ0r6 = -184.9052456
Const BJ0s1 = 57568490411#
Const BJ0s2 = 1029532985
Const BJ0s3 = 9494680.718
Const BJ0s4 = 59272.64853
Const BJ0s5 = 267.8532712
Const BJ0s6 = 1

Const BJ1p1 = 1
Const BJ1p2 = .00183105
Const BJ1p3 = -.00003516396496
Const BJ1p4 = .000002457520174
Const BJ1p5 = -.000000240337019
Const BJ1q1 = .04687499995
Const BJ1q2 = -.0002002690873
Const BJ1q3 = .000008449199096
Const BJ1q4 = -.00000088228987
Const BJ1q5 = .000000105787412
Const BJ1r1 = 72362614232#
Const BJ1r2 = -7895059235#
Const BJ1r3 = 242396853.1
Const BJ1r4 = -2972611.439
Const BJ1r5 = 15704.4826
Const BJ1r6 = -30.16036606
Const BJ1s1 = 144725228442#
Const BJ1s2 = 2300535178#
Const BJ1s3 = 18583304.74
Const BJ1s4 = 99447.43394
Const BJ1s5 = 376.9991397
Const BJ1s6 = 1

Sub AnalMain ()
'NOTE:  Height is actually half height of inf. slab.

   Pi = 4 * Atn(1)
   Call MachPrec

      BiotCyl = hConvection * Radius / kT
      BiotSlab = hConvection * Height / kT

      NumRoots = 10
    Call AnalSolutionLoop

End Sub

Sub AnalSolutionLoop ()

ReDim TempAtrxt(11) As Double, r(11) As Double

   tPosition = SolutionTime
   frmOutPut.txtTime = SolutionTime / 60
   For i = 1 To 11
      Select Case i
         Case 1
            rPosition = 0
            xPosition = 0
         Case 2
            rPosition = .1 * Radius
            xPosition = 0
         Case 3
            rPosition = .2 * Radius
            xPosition = 0
      Case 4
        rPosition = .3 * Radius
        xPosition = 0
      Case 5
        rPosition = .4 * Radius
        xPosition = 0
      Case 6
        rPosition = .5 * Radius
        xPosition = 0
      Case 7
        rPosition = .6 * Radius
        xPosition = 0
      Case 8
        rPosition = .7 * Radius
        xPosition = 0
      Case 9
        rPosition = .8 * Radius
        xPosition = 0
      Case 10
        rPosition = .9 * Radius
        xPosition = 0
      Case 11
        rPosition = Radius
        xPosition = 0
    End Select
    r(i) = rPosition

    Call InfCyl

    Call InfSlab

    TempAtrxt(i) = AmbientTemp + (InitialTemp - AmbientTemp) * thetaCyl *
thetaSlab
  Next i


'  OutPut Solution to Grid

StartAtRow = StartAtRow + 3
frmOutPut.grdOutPut.Row = StartAtRow
For i = 1 To 11
frmOutPut.grdOutPut.Col = i
  frmOutPut.grdOutPut.Text = Format(TempAtrxt(i), "Scientific")
Next i
```

```
frmOutPut.Refresh

End Sub

Function BessJ0 (x As Double) As Double
' Numerical Recipies p.(225)
' Returns the Bessel fn J0(x) for any real x

Dim xx As Double, ax As Double, y As Double, z As Double

If Abs(x) < 8 Then
   y = x * x
   BessJ0 = (BJ0r1 + y * (BJ0r2 + y * (BJ0r3 + y * (BJ0r4 + y * (BJ0r5 + y *
BJ0r6))))) / (BJ0s1 + y * (BJ0s2 + y * (BJ0s3 + y * (BJ0s4 + y * (BJ0s5 + y *
BJ0s6)))))
Else
   ax = Abs(x)
   z = 8 / ax
   y = z * z
   xx = ax - .785398164
   BessJ0 = Sqr(.636619772 / ax) * (Cos(xx) * (BJ0p1 + y * (BJ0p2 + y * (BJ0p3 + y
* (BJ0p4 + y * BJ0p5)))) - z * Sin(xx) * (BJ0q1 + y * (BJ0q2 + y * (BJ0q3 + y *
(BJ0q4 + y * BJ0q5)))))
End If

End Function

Function BessJ1 (x As Double) As Double
' Numerical Recipies p.226
' Returns the Bessel fn J1(x) for any real x

Dim xx As Double, ax As Double, y As Double, z As Double


If Abs(x) < 8 Then
   y = x * x
   BessJ1 = x * (BJ1r1 + y * (BJ1r2 + y * (BJ1r3 + y * (BJ1r4 + y * (BJ1r5 + y *
BJ1r6))))) / (BJ1s1 + y * (BJ1s2 + y * (BJ1s3 + y * (BJ1s4 + y * (BJ1s5 + y *
BJ1s6)))))
Else
   ax = Abs(x)
   z = 8 / ax
   y = z * z
```

```
  xx = ax - 2.356194491
  BessJ1 = Sqr(.636619772 / ax) * (Cos(xx) * (BJ1p1 + y * (BJ1p2 + y * (BJ1p3 + y
* (BJ1p4 + y * BJ1p5)))) - z * Sin(xx) * (BJ1q1 + y * (BJ1q2 + y * (BJ1q3 + y *
(BJ1q4 + y * BJ1q5)))))
End If

End Function

Function CylFunc (x As Double, Bi As Double) As Double

  CylFunc = Bi * BessJ0(x) - x * BessJ1(x)

End Function

Sub FindTransRoot (Geometry As String)
' Find Roots of transcendental functions for heat Transfer

Dim fmid  As Double, F As Double
Dim MinRoot As Double, MaxRoot As Double, RootTol As Double
Dim Biot As Double, RootIndex As Integer

' Set bounds for search of roots via Bisection (rtbis)
Select Case Geometry
  Case "Slab"
    MinRoot = 0
    MaxRoot = Pi / 2
    Biot = BiotSlab
  Case "Cylinder"
    MinRoot = 0
    MaxRoot = Pi
    Biot = BiotCyl
End Select

RootTol = .00001

For RootIndex = 1 To NumRoots

  TransRoot(RootIndex) = rtbis(MinRoot, MaxRoot, RootTol, Geometry, Biot)
  If TransRoot(RootIndex) < RootTol Then TransRoot(RootIndex) = 0

  Select Case Geometry
    Case "Slab"
      MinRoot = MinRoot + Pi
      MaxRoot = TransRoot(RootIndex) + Pi
```

```
        Case "Cylinder"
           MinRoot = MinRoot + Pi
           MaxRoot = MaxRoot + Pi
     End Select
 Next RootIndex

 End Sub

 Sub InfCyl ()

 ReDim TransRoot(NumRoots) As Double
 Dim RootIndex As Integer

 Dim TermConstant As Double, TempTerm As Double
 Dim BJ0 As Double, BJ1 As Double


 Call FindTransRoot("Cylinder")

thetaCyl = 0
   For RootIndex = 1 To NumRoots
      If TransRoot(RootIndex) = 0 Then TransRoot(RootIndex) =
TransRoot(RootIndex) + eps

         BJ0 = BessJ0(TransRoot(RootIndex))
         BJ1 = BessJ1(TransRoot(RootIndex))

         TermConstant = 2 * BJ1 / (TransRoot(RootIndex) * (BJ0 * BJ0 + BJ1 * BJ1))
         TempTerm = (TermConstant * BessJ0(TransRoot(RootIndex) * rPosition /
Radius)) * Exp(-TransRoot(RootIndex) * TransRoot(RootIndex) * Alpha * tPosition /
(Radius * Radius))

         If Abs(TempTerm * (InitialTemp - AmbientTemp)) <= .001 Then
            thetaCyl = thetaCyl + TempTerm
            Exit For
         End If

         thetaCyl = thetaCyl + TempTerm
  Next RootIndex

End Sub

Sub InfSlab ()
' Analytical solution - Slab (B.C. third kind)
```

```
ReDim TransRoot(NumRoots) As Double
Dim SumTemp As Double, TempTermConstant As Double, TempTerm As Double
Dim RootIndex As Integer


Call FindTransRoot("Slab")

thetaSlab = 0
  For RootIndex = 1 To NumRoots
    If TransRoot(RootIndex) > 0 Then
      TempTermConstant = 2 * Sin(TransRoot(RootIndex)) /
(TransRoot(RootIndex) + Sin(TransRoot(RootIndex)) * Cos(TransRoot(RootIndex)))
    Else
      TempTermConstant = 1
    End If

    TempTerm = (TempTermConstant * Cos(TransRoot(RootIndex) * xPosition /
Height)) * Exp(-TransRoot(RootIndex) * TransRoot(RootIndex) * Alpha * tPosition /
(Height * Height))

    thetaSlab = thetaSlab + TempTerm

    If Abs(TempTerm * (InitialTemp - AmbientTemp)) <= .001 Then
      Exit For
    End If

  Next RootIndex

End Sub

Sub MachPrec ()

eps = 1
Do
  eps = eps / 2
Loop While 1 + eps > 1
eps = eps * 2


End Sub

Function rtbis (MinU As Double, MaxU As Double, Tol As Double, Geometry As
String, Biot As Double)
```

```
' Bisection roots finder
Dim p As Double, Fp As Double, Fx1 As Double, Fx2 As Double
Dim x1 As Double, x2 As Double

x1 = MinU
x2 = MaxU
Do
   p = x1 + (x2 - x1) / 2
   Select Case Geometry
      Case "Slab"
         Fp = SlabFunc(p, Biot)
         Fx1 = SlabFunc(x1, Biot)
         Fx2 = SlabFunc(x2, Biot)
      Case "Cylinder"
         Fp = CylFunc(p, Biot)
         Fx1 = CylFunc(x1, Biot)
         Fx2 = CylFunc(x2, Biot)
   End Select

   If Fp = 0 Or (x2 - x1) / 2 < Tol Then
      rtbis = p
      Exit Function
   Else
      If Fx1 = 0 Then
         If Fp * Fx2 > 0 Then
            x2 = p
         Else
            x1 = p
         End If
      Else
         If Fp * Fx1 > 0 Then
            x1 = p
         Else
            x2 = p
         End If
      End If
   End If
Loop

End Function
Function SlabFunc (x As Double, Bi As Double) As Double
   SlabFunc = x * Tan(x) - Bi
End Function
```

## APPENDIX C
## PEIE CODE FOR CONDUCTION HEATING CANNED FOOD

```
‘MODULE 1 of 2:  PEIE for conduction heating canned foods.
‘
‘Modeule 1 provides input functions and then launches the PEIE code
‘by calling subroutine NOCAPSFC in module 2.

Option Explicit

Global NodesDim1 As Integer, NodesDim2 As Integer, NodesDim3 As Integer

Global Dim1 As Double, Dim2 As Double, Dim3 As Double

Global T0 As Double, TaH As Double, TaC As Double
Global Co As Double
Global TaHTime As Single, TaCTime As Single
Global T() As Double, TempEq() As Double, TimeEQ() As Double

Global Cp As Double, kT As Double, Density As Double
Global Alpha As Double
Global hHeat As Double, hCool As Double
Global hIceWater As Double, hAir As Double

Global InFileName() As String, OutFileName As String, NumSelected  As Integer
Global FileName1 As String, FileName2 As String
Dim SelectedArray() As Integer
Dim NumFilePairs As Integer, PairCount As Integer


Global Ea1 As Double, Ea2 As Double
Global IsBadFilePair As Integer

Global Const Rg = 8.31441: '    J/(mol K)

Sub CheckFilePairs ()
```

```
Dim i As Integer, j As Integer, IsLastBackSlash As Integer
Dim file1 As String, file2 As String
Dim TempCode1 As String, TempCode2 As String: ' TempCode stands for
temperature code of the experiment

      PairCount = 0
      For i = 1 To NumSelected - 1
          InFileName(1) = frmFileSelect.dir1.Path & "\" &
frmFileSelect.fil1.List(SelectedArray(i))

          file1 = frmFileSelect.fil1.List(SelectedArray(i))
          FileName1 = file1
          file1 = Left$(file1, 5)


          For j = i + 1 To NumSelected
              InFileName(2) = frmFileSelect.dir1.Path & "\" &
frmFileSelect.fil1.List(SelectedArray(j))
              file2 = frmFileSelect.fil1.List(SelectedArray(j))
              FileName2 = file2
              file2 = Left$(file2, 5)

                PairCount = PairCount + 1
                If file1 = file2 Then IsBadFilePair = True

                OutEPM1.txtPairNumber.Text = PairCount
                OutEPM1.lstCurFiles.Clear
                OutEPM1.lstCurFiles.AddItem frmFileSelect.fil1.List(SelectedArray(i))
                OutEPM1.lstCurFiles.AddItem frmFileSelect.fil1.List(SelectedArray(j))

                Ea1 = Val(InEPM1.txtEa1.Text)
                Ea2 = Val(InEPM1.txtEa2.Text)

             If IsBadFilePair = False Then
                Call NoCapCyl
             Else
                IsBadFilePair = False
             End If
         Next j
      Next i

End Sub

Sub FileSelect ()
```

```
Dim i As Integer

   NumSelected = 0
   For i = 0 To frmFileSelect.fil1.ListCount - 1
      If frmFileSelect.fil1.Selected(i) Then

         NumSelected = NumSelected + 1

         ReDim Preserve SelectedArray(NumSelected)

         SelectedArray(NumSelected) = i

      End If
   Next i

NumFilePairs = NumSelected * (NumSelected - 1) / 2


End Sub

Sub Main ()
ReDim InFileName(2)

IsBadFilePair = False

   Call SetInputVariables

   If InEPM1.opt3dFinCyl Then

      Call FileSelect
         OutEPM1.txtNumFilesSelected.Text = NumSelected

         If NumSelected = 1 Then
            InFileName(1) = frmFileSelect.dir1.Path & "\" &
frmFileSelect.fil1.List(SelectedArray(1))
            Call NoCapCyl
         Else
            OutEPM1.txtTotNumPairs.Text = NumFilePairs
            Call CheckFilePairs
         End If
   End If

End Sub
```

```
Sub SetInputVariables ()
Dim i  As Integer

   OutFileName = frmOutFileSelect.dir1.Path & "\" &
frmOutFileSelect.txtOutFile.Text

   Dim1 = Val(InEPM1.txtDim1.Text) / 2
   Dim2 = Val(InEPM1.txtDim2.Text) / 2
   Dim3 = Val(InEPM1.txtDim3.Text) / 2
      If InEPM1.optInches Then
         Dim1 = Dim1 * 2.54
         Dim2 = Dim2 * 2.54
         Dim3 = Dim3 * 2.54
      End If

   NodesDim1 = Val(InEPM1.txtNodesDim1.Text)
   NodesDim2 = Val(InEPM1.txtNodesDim2.Text)
   NodesDim3 = Val(InEPM1.txtNodesDim3.Text)

      T0 = Val(InEPM1.txtT0.Text)
      Co = Val(InEPM1.txtCo.Text)

   kT = Val(InEPM1.txtK.Text)
   Cp = Val(InEPM1.txtCp.Text)
   Density = Val(InEPM1.txtDensity.Text)

   Alpha = kT / (Density * Cp):  'Val(InEPM1.txtAlpha.Text)

   hHeat = Val(InEPM1.txthHeat.Text)
   hCool = Val(InEPM1.txthCool.Text)
   hAir = Val(InEPM1.txthAir.Text)
   hIceWater = Val(InEPM1.txthIceWater.Text)

   Ea1 = Val(InEPM1.txtEa1)
   Ea2 = Val(InEPM1.txtEa2)

End Sub
```

```
‘MODULE 2 of 2

Option Explicit

Dim i As Integer, j As Integer, k As Integer
Dim r() As Double
Dim dr() As Double, rA() As Double
Dim dy() As Double, yA() As Double
Dim dVol() As Double, FCylVol As Double
Dim TimeStep As Double
Dim tNow As Double
Dim Nr As Integer, Ny As Integer
Dim Radius As Double, Height As Double
Dim hUse As Double, Tamb As Double
Dim Told() As Double
Dim RetortMode As String * 1, RMode() As String * 1
Dim RetortTime As Double, RetortTemp As Double
Dim RTime() As Double, RTemp() As Double
Dim RetortDataIndex As Integer, NumRetortData As Integer
Dim FinalMassAvgConc() As Double
Dim GraphIndex As Integer
Dim GraphNode() As Single
Dim Gs1() As Double, Gs2() As Double
Dim ConcRatio1() As Double, ConcRatio2() As Double
Dim iso_k1() As Double, iso_k2() As Double
Dim eqTemp() As Double, eqTime() As Double
Dim Kref1  As Double, Kref2 As Double

Sub DisplayData (LoopCount As Integer)

   Call Print2Grid
   If LoopCount Mod 100 = 0 Then
      GraphIndex = GraphIndex + 1
      ReDim Preserve GraphNode(2, GraphIndex)
         GraphNode(1, GraphIndex) = T(1, 1): '  Center of Can
         GraphNode(2, GraphIndex) = Tamb: '    RetortTemp
      If GraphIndex > 1 Then Call Print2Graph2
      Call Print2Graph1
      LoopCount = 1
   End If

End Sub

Private Sub FCylSimulationLoop ()
```

```
Dim LoopCount As Integer
Dim TermCoef As Double
Dim a1 As Double, a2 As Double, a3 As Double, a4 As Double
Dim UAr As Double, UAy As Double
Dim AvgNodeTemp As Double

tNow = 0
LoopCount = 0
GraphIndex = 0
RetortDataIndex = 1

Do

tNow = tNow + TimeStep


   Select Case tNow
      Case Is >= RTime(NumRetortData)
         Tamb = 0
         hUse = hIceWater
         If T(1, 1) < 30 Then
            Call DisplayData(LoopCount)
            Exit Do: '              exit when Tcenter < 10 C.
         End If
      Case Else
         Tamb = GetAmbientTemp()
         hUse = GetConvection()
  End Select

'  Check if time to display data
   If LoopCount Mod 100 = 0 Then
      Call DisplayData(LoopCount)
   End If
   LoopCount = LoopCount + 1

For j = Ny - 1 To 1 Step -1
   For i = Nr - 1 To 1 Step -1
   Select Case j
      Case 1
         TermCoef = Alpha * TimeStep / dVol(i, j)
         Select Case i
            Case 1
               a1 = rA(i, j) / dr(i)
               a2 = yA(i) / dy(j)
               T(i, j) = Told(i, j) * (1 - TermCoef * (a1 + a2)) + TermCoef * (a1 *
Told(i + 1, j) + a2 * Told(i, j + 1))
            Case 2 To Nr - 2
               a1 = rA(i - 1, j) / dr(i - 1)
               a2 = rA(i, j) / dr(i)
               a3 = yA(i) / dy(j)
               T(i, j) = Told(i, j) * (1 - TermCoef * (a1 + a2 + a3)) + TermCoef * (a1 *
Told(i - 1, 1) + a2 * Told(i + 1, 1) + a3 * Told(i, j + 1))
            Case Nr - 1
               a1 = rA(i - 1, j) / dr(i - 1)
               a2 = yA(i) / dy(j)
               UAr = 1 / (kT / (hUse * rA(i + 1, j)) + dr(i) / rA(i, j))
               T(i, j) = Told(i, j) * (1 - TermCoef * (a1 + a2 + UAr)) + TermCoef * (a1
* Told(i - 1, j) + a2 * Told(i, j + 1) + Tamb * UAr)
               'T(i + 1, j) = (kT * rA(i, j) * T(i, j) / dr(i) + hUse * rA(i + 1, j) * Tamb) /
(hUse * rA(i + 1, j) + kT * rA(i, j) / dr(i))
         End Select
      Case 2 To Ny - 2
         TermCoef = Alpha * TimeStep / dVol(i, j)
         Select Case i
            Case 1
               a1 = rA(i, j) / dr(i)
               a2 = yA(i) / dy(j - 1)
               a3 = yA(i) / dy(j)
               T(i, j) = Told(i, j) * (1 - TermCoef * (a1 + a2 + a3)) + TermCoef * (a1 *
Told(i + 1, j) + a2 * Told(i, j - 1) + a3 * Told(i, j + 1))
            Case 2 To Nr - 2
               a1 = rA(i - 1, j) / dr(i)
               a2 = rA(i, j) / dr(i)
               a3 = yA(i) / dy(j - 1)
               a4 = yA(i) / dy(j)
               T(i, j) = Told(i, j) * (1 - TermCoef * (a1 + a2 + a3 + a4)) + TermCoef *
(a1 * Told(i - 1, j) + a2 * Told(i + 1, j) + a3 * Told(i, j - 1) + a4 * Told(i, j + 1))
            Case Nr - 1
               a1 = rA(i - 1, j) / dr(i - 1)
               a2 = yA(i) / dy(j - 1)
               a3 = yA(i) / dy(j)
               UAr = 1 / (kT / (hUse * rA(i + 1, j)) + dr(i) / rA(i, j))
               T(i, j) = Told(i, j) * (1 - TermCoef * (a1 + a2 + a3 + UAr)) + TermCoef *
(a1 * Told(i - 1, j) + a2 * Told(i, j - 1) + a3 * Told(i, j + 1) + Tamb * UAr)
               'T(i + 1, j) = (kT * rA(i, j) * T(i, j) / dr(i) + hUse * rA(i + 1, j) * Tamb) /
(hUse * rA(i + 1, j) + kT * rA(i, j) / dr(i))
         End Select
```

```
     Case Ny - 1
       TermCoef = Alpha * TimeStep / dVol(i, j)
       Select Case i
         Case 1
           a1 = rA(i, j) / dr(i)
           a2 = yA(i) / dy(j - 1)
           UAy = 1 / (kT / (hUse * yA(i)) + dy(j) / yA(i))
           T(i, j) = Told(i, j) * (1 - TermCoef * (a1 + a2 + UAy)) + TermCoef * (a1
* Told(i + 1, j) + a2 * Told(i, j - 1) + Tamb * UAy)
           'T(i, j + 1) = (kT * T(i, j) / dy(j) + hUse * Tamb) / (hUse + kT / dy(j))
         Case 2 To Nr - 2
           a1 = rA(i - 1, j) / dr(i - 1)
           a2 = rA(i, j) / dr(i)
           a3 = yA(i) / dy(j - 1)
           UAy = 1 / (kT / (hUse * yA(i)) + dy(j) / yA(i))
           T(i, j) = Told(i, j) * (1 - TermCoef * (a1 + a2 + a3 + UAy)) + TermCoef *
(a1 * Told(i - 1, j) + a2 * Told(i + 1, j) + a3 * Told(i, j - 1) + Tamb * UAy)
           'T(i, j + 1) = (kT * T(i, j) / dy(j) + hUse * Tamb) / (hUse + kT / dy(j))
         Case Nr - 1
           a1 = rA(i - 1, j) / dr(i - 1)
           a2 = yA(i) / dy(j - 1)
           UAy = 1 / (kT / (hUse * yA(i)) + dy(j) / yA(i))
           UAr = 1 / (kT / (hUse * rA(i + 1, j)) + dr(i) / rA(i, j))
           T(i, j) = Told(i, j) * (1 - TermCoef * (a1 + a2 + UAy + UAr)) +
TermCoef * (a1 * Told(i - 1, j) + a2 * Told(i, j - 1) + Tamb * (UAy + UAr))
           'T(i, j + 1) = (kT * T(i, j) / dy(j) + hUse * Tamb) / (hUse + kT / dy(j))
           'T(i + 1, j) = (kT * rA(i, j) * T(i, j) / dr(i) + hUse * rA(i + 1, j) * Tamb) /
(hUse * rA(i + 1, j) + kT * rA(i, j) / dr(i))
           'T(Nr, Ny) = (T(i, j + 1) + T(i + 1, j)) / 2
      End Select
    End Select
  Call GsAccumulate
Next i
Next j

DoEvents

For j = 1 To Ny - 1
  For i = 1 To Nr - 1
      Told(i, j) = T(i, j)
  Next i
Next j

Loop
```

```
End Sub

Private Function GetAmbientTemp ()

Dim LinearInterp As Double, TempDiff As Double

Select Case tNow
   Case Is >= RTime(RetortDataIndex + 1)
      Do While tNow > RTime(RetortDataIndex + 1)
         RetortDataIndex = RetortDataIndex + 1
      Loop
      LinearInterp = (tNow - RTime(RetortDataIndex)) / (RTime(RetortDataIndex + 1)
- RTime(RetortDataIndex))
      TempDiff = RTemp(RetortDataIndex + 1) - RTemp(RetortDataIndex)
      GetAmbientTemp = RTemp(RetortDataIndex) + LinearInterp * TempDiff
   Case Else
      LinearInterp = (tNow - RTime(RetortDataIndex)) / (RTime(RetortDataIndex + 1)
- RTime(RetortDataIndex))
      TempDiff = RTemp(RetortDataIndex + 1) - RTemp(RetortDataIndex)
      GetAmbientTemp = RTemp(RetortDataIndex) + LinearInterp * TempDiff
End Select

End Function

Function GetConvection () As Double

   Select Case RMode(RetortDataIndex)
      Case "a"
         GetConvection = hAir
      Case "h"
         GetConvection = hHeat
      Case "c"
         GetConvection = hCool
   End Select

End Function

Private Sub GetRetortData (FileIndex As Integer)
Dim i As Integer, crap As String, Dummy As Single
Dim d0_MassAvgConc As Double, d1_MassAvgConc As Double, d2_MassAvgConc
As Double

Dim d0_L As Double, d1_L As Double, d2_L As Double
Dim d0_a As Double, d1_a As Double, d2_a As Double
Dim d0_b As Double, d1_b As Double, d2_b As Double

Dim d0_Lh As Double, d1_Lh As Double, d2_Lh As Double
Dim d0_ah As Double, d1_ah As Double, d2_ah As Double
Dim d0_bh As Double, d1_bh As Double, d2_bh As Double

    Open InFileName(FileIndex) For Input As #1

    Input #1, d0_MassAvgConc, Dummy
    Input #1, d1_MassAvgConc, Dummy
    Input #1, d2_MassAvgConc, Dummy
    Input #1, d0_L, d0_a, d0_b, d0_Lh, d0_ah, d0_bh
    Input #1, d1_L, d1_a, d1_b, d1_Lh, d1_ah, d1_bh
    Input #1, d2_L, d2_a, d2_b, d2_Lh, d2_ah, d2_bh
    Line Input #1, crap

    i = 0
    Do While Not EOF(1)
        i = i + 1
        ReDim Preserve RTime(i), RTemp(i), RMode(i)
        Input #1, RetortMode, RetortTime, RetortTemp, Dummy, Dummy, Dummy
        RTime(i) = RetortTime * 60
        RTemp(i) = RetortTemp
        RMode(i) = RetortMode
    Loop

    Close #1
    NumRetortData = i

    If InEpm1.optBugs_d0 Then
        FinalMassAvgConc(FileIndex) = d0_MassAvgConc
        If FinalMassAvgConc(FileIndex) = 0 Then IsBadFilePair = True
    ElseIf InEpm1.optBugs_d1 Then
        FinalMassAvgConc(FileIndex) = d1_MassAvgConc
        If FinalMassAvgConc(FileIndex) = 0 Then IsBadFilePair = True
    ElseIf InEpm1.optBugs_d2 Then
        FinalMassAvgConc(FileIndex) = d2_MassAvgConc
        If FinalMassAvgConc(FileIndex) = 0 Then IsBadFilePair = True
    ElseIf InEpm1.optColor_0 Then
        FinalMassAvgConc(FileIndex) = -d0_Lh * d0_ah / d0_bh
        If FinalMassAvgConc(FileIndex) = 0 Or d0_ah >= 0 Then IsBadFilePair = True
    ElseIf InEpm1.optColor_1 Then
        FinalMassAvgConc(FileIndex) = -d1_Lh * d1_ah / d1_bh
```

```
      If FinalMassAvgConc(FileIndex) = 0 Or d1_ah >= 0 Then IsBadFilePair = True
   ElseIf InEpm1.optColor_2 Then
      FinalMassAvgConc(FileIndex) = -d2_Lh * d2_ah / d2_bh
      If FinalMassAvgConc(FileIndex) = 0 Or d2_ah >= 0 Then IsBadFilePair = True
   End If

End Sub

Sub GsAccumulate ()
Dim Z As Double, rxnTemp As Double

'  calc using 55% of temp change
   If T(i, j) > Told(i, j) Then
      rxnTemp = Told(i, j) + .55 * (T(i, j) - Told(i, j))
   Else
      rxnTemp = T(i, j) + .55 * (Told(i, j) - T(i, j))
   End If

   Z = ((121.1 + 273.15) - (rxnTemp + 273.15)) / ((121.1 + 273.15) * (rxnTemp +
273.15))

   Gs1(i, j) = Gs1(i, j) + Exp(-Ea1 * Z / Rg)
   Gs2(i, j) = Gs2(i, j) + Exp(-Ea2 * Z / Rg)

End Sub

Sub Isothermal (FileIndex As Integer)
Dim CheckSum As Double

For j = 1 To Ny - 1
   For i = 1 To Nr - 1
      Gs1(i, j) = Gs1(i, j) * TimeStep / 60
      Gs2(i, j) = Gs2(i, j) * TimeStep / 60
      If Gs1(i, j) = Gs2(i, j) Then Gs2(i, j) = Gs2(i, j) - .000001
   Next i
Next j

   Kref1 = RootBisection(FileIndex, Gs1(), 0, 1000, .000000000001)
   Kref2 = RootBisection(FileIndex, Gs2(), 0, 1000, .000000000001)
If IsBadFilePair = True Then Exit Sub


For j = 1 To Ny - 1
   For i = 1 To Nr - 1
```

```
'      C/Co = 1/exp(Kref*Gs).
       ConcRatio1(i, j) = Exp(-Kref1 * Gs1(i, j))
       ConcRatio2(i, j) = Exp(-Kref2 * Gs2(i, j))
       If ConcRatio1(i, j) = 0 Or ConcRatio2(i, j) = 0 Then
          IsBadFilePair = True
          Exit Sub
       End If

'      CheckSum = CheckSum + ConcRatio1(i, j) * Co * dVol(i, j) / FCylVol: 'should
equal tot num bugs left in can

       eqTemp(FileIndex, i, j) = 1 / (Rg * Log(Gs1(i, j) / Gs2(i, j)) / (Ea2 - Ea1) + 1 /
(121.1 + 273.15))
       eqTime(FileIndex, i, j) = Gs1(i, j) * Exp(Ea1 / Rg * ((121.1 + 273.15) -
eqTemp(FileIndex, i, j)) / ((121.1 + 273.15) * eqTemp(FileIndex, i, j)))

       iso_k1(FileIndex, i, j) = -Log(ConcRatio1(i, j)) / eqTime(FileIndex, i, j)
       iso_k2(FileIndex, i, j) = -Log(ConcRatio2(i, j)) / eqTime(FileIndex, i, j)
   Next i
Next j


End Sub

Function Kref_Gs_Func (FileIndex As Integer, Gs() As Double, x As Double) As
Double
Dim Dummy As Double
ReDim Preserve Gs(Nr - 1, Ny - 1)

For j = 1 To Ny - 1
   For i = 1 To Nr - 1
        Dummy = Dummy + dVol(i, j) * Exp(-x * Gs(i, j))
   Next i
Next j

   Kref_Gs_Func = FinalMassAvgConc(FileIndex) * FCylVol / Co - Dummy

End Function

Private Function MinFCylTimeStep () As Double

Dim i As Integer, j As Integer
Dim MinTime As Double, dtSmallest As Double
Dim UA As Double, UAr As Double, UAy As Double
Dim a1 As Double, a2 As Double, a3 As Double, a4 As Double
ReDim dt(9) As Double

'  bot left (can center)
   i = 1: j = 1
   a1 = rA(i, j) / dr(i)
   a2 = yA(i) / dy(j)
   dt(1) = dVol(i, j) / (Alpha * (a1 + a2))

'  bot right
   i = Nr - 1: j = 1
   a1 = rA(i - 1, j) / dr(i - 1)
   a2 = yA(i) / dy(j)
   UA = 1 / (kT / (hHeat * rA(i + 1, j)) + dr(i) / rA(i, j))
   dt(2) = dVol(i, j) / (Alpha * (a1 + a2 + UA))

'  top left (Can top center)
   i = 1: j = Ny - 1
   a1 = rA(i, j + 1) / dr(i)
   a2 = yA(i) / dy(j - 1)
   UA = 1 / (kT / (hHeat * yA(i)) + dy(j) / yA(i))
   dt(3) = dVol(i, j) / (Alpha * (a1 + a2 + UA))

'  top right
   i = Nr - 1: j = Ny - 1
   a1 = rA(i - 1, j + 1) / dr(i - 1)
   a2 = yA(i) / dy(j - 1)
   UAy = 1 / (kT / (hHeat * yA(i)) + dy(j) / yA(i))
   UAr = 1 / (kT / (hHeat * rA(i + 1, j)) + dr(i) / rA(i, j))
   dt(4) = dVol(i, j) / (Alpha * (a1 + a2 + UAy + UAr))

'  bot interior
   j = 1
   a1 = rA(1, j) / dr(1)
   a2 = rA(2, j) / dr(2)
   a3 = yA(2) / dy(j)
   dt(5) = dVol(2, j) / (Alpha * (a1 + a2 + a3))
   For i = 3 To Nr - 2
      a1 = rA(i - 1, j) / dr(i - 1)
      a2 = rA(i, j) / dr(i)
      a3 = yA(i) / dy(j)
      MinTime = dVol(i, j) / (Alpha * (a1 + a2 + a3))
      If MinTime < dt(5) Then dt(5) = MinTime
```

```
    Next i

'   vertical center line - interior
    i = 1: j = 2
    a1 = rA(i, j) / dr(i)
    a2 = yA(i) / dy(j)
    dt(6) = dVol(i, j) / (Alpha * (a1 + 2 * a2))

'   Side surface - interior
    i = Nr - 1: j = 2
    a1 = rA(i - 1, j) / dr(i - 1)
    a2 = yA(i) / dy(j)
    UA = 1 / (kT / (hHeat * rA(i + 1, j)) + dr(i) / rA(i, j))
    dt(7) = dVol(i, j) / (Alpha * (a1 + 2 * a2 + UA))

'   Top surface - interior
    j = Ny - 1
    a1 = rA(1, j) / dr(1)
    a2 = rA(2, j) / dr(2)
    a3 = yA(2) / dy(j - 1)
    UA = 1 / (kT / (hHeat * yA(2)) + dy(j) / yA(2))
    dt(8) = dVol(2, j) / (Alpha * (a1 + a2 + a3 + UA))
    For i = 3 To Nr - 2
       a1 = rA(i - 1, j) / dr(i - 1)
       a2 = rA(i, j) / dr(i)
       a3 = yA(i) / dy(j - 1)
       UA = 1 / (kT / (hHeat * yA(i)) + dy(j) / yA(i))
       MinTime = dVol(i, j) / (Alpha * (a1 + a2 + a3 + UA))
       If MinTime < dt(8) Then dt(8) = MinTime
    Next i

'   Interior nodes
    j = 3
    a1 = rA(1, j) / dr(1)
    a2 = rA(2, j) / dr(2)
    a3 = yA(1) / dy(j)
    a4 = yA(1) / dy(j - 1)
    dt(9) = dVol(2, j) / (Alpha * (a1 + a2 + a3 + a4))
    For i = 3 To Nr - 2
       a1 = rA(i - 1, j) / dr(i - 1)
       a2 = rA(i, j) / dr(i)
       a3 = yA(i) / dy(j)
       a4 = yA(i) / dy(j - 1)
       MinTime = dVol(i, j) / (Alpha * (a1 + a2 + a3 + a4))
```

```
        If MinTime < dt(9) Then dt(9) = MinTime
    Next i

    dtSmallest = dt(1)
    For i = 2 To 9
        If dt(i) < dtSmallest Then dtSmallest = dt(i)
    Next i

    MinFCylTimeStep = dtSmallest / 2

End Function

Sub NewEaRange (MinEa As Double, MaxEa As Double)
Dim dEqTemp As Double
Dim rBestIndex As Integer, yBestIndex As Integer, i As Integer, j As Integer
Dim MaxTeqDiff As Double, TestTeqDiff As Double

rBestIndex = 1
yBestIndex = 1
MaxTeqDiff = Abs(eqTemp(1, 1, 1) - eqTemp(2, 1, 1))
For j = 2 To Ny - 1
    For i = 2 To Nr - 1
        TestTeqDiff = Abs(eqTemp(1, i, j) - eqTemp(2, i, j))
        If TestTeqDiff > MaxTeqDiff Then
            MaxTeqDiff = TestTeqDiff
            rBestIndex = i
            yBestIndex = j
        End If
    Next i
Next j

dEqTemp = eqTemp(1, rBestIndex, yBestIndex) - eqTemp(2, rBestIndex, yBestIndex)
If Abs(dEqTemp) < .001 Then
    Stop
Else
    dEqTemp = dEqTemp / (eqTemp(1, rBestIndex, yBestIndex) * eqTemp(2,
rBestIndex, yBestIndex))
    MinEa = Rg * Log(iso_k1(1, rBestIndex, yBestIndex) / iso_k1(2, rBestIndex,
yBestIndex)) / dEqTemp
    MaxEa = Rg * Log(iso_k2(1, rBestIndex, yBestIndex) / iso_k2(2, rBestIndex,
yBestIndex)) / ((eqTemp(1, rBestIndex, yBestIndex) - eqTemp(2, rBestIndex,
yBestIndex)) / (eqTemp(1, rBestIndex, yBestIndex) * eqTemp(2, rBestIndex,
yBestIndex)))
```

```
End If

End Sub

Sub NoCapCyl ()

Dim i As Integer, j As Integer, k As Integer
Dim EaModCount As Integer
Dim MaxEaOld As Double, MinEaOld As Double
Dim MaxEa As Double, MinEa As Double, AvgEa As Double, AvgKref As Double
Dim Z As Double

Nr = NodesDim1
Ny = NodesDim2
Radius = Dim1
Height = Dim2

IsBadFilePair = False

   If Ea2 > Ea1 Then
      MaxEaOld = Ea2
      MinEaOld = Ea1
   Else
      MaxEaOld = Ea1
      MinEaOld = Ea2
   End If

   Call SetFCylDimensions

   TimeStep = MinFCylTimeStep()

EaModCount = 0
Do
   ReDim eqTemp(2, Nr - 1, Ny - 1), eqTime(2, Nr - 1, Ny - 1)
   ReDim iso_k1(2, Nr - 1, Ny - 1), iso_k2(2, Nr - 1, Ny - 1)
   ReDim FinalMassAvgConc(2)

      EaModCount = EaModCount + 1
      OutEPM1.txtLoopCount.Text = EaModCount
      OutEPM1.mskEa1.Mask = Ea1
      OutEPM1.mskEa2.Mask = Ea2

  For i = 1 To 2: '     NumSelected
```

```
      Call GetRetortData(i)
      If IsBadFilePair Then Exit Do

      Call SetFCylInitialTemp

      Call FCylSimulationLoop

      Call Isothermal(i)
      If IsBadFilePair Then Exit Do
   Next i

   Call NewEaRange(MinEa, MaxEa)

   If MaxEa < 0 Or MinEa < 0 Then
      IsBadFilePair = True
      Exit Do
   ElseIf Abs(MinEa - MaxEa) < 50 And Abs(MinEa - MinEaOld) < 25 And
Abs(MaxEa - MaxEaOld) < 25 Then
      Exit Do
   ElseIf EaModCount >= 100 Then
      IsBadFilePair = True
      Exit Do
   Else
      If Ea1 = Ea2 Then Ea1 = Ea1 - .000001
      Ea1 = MinEa
      MinEaOld = Ea1
      Ea2 = MaxEa
      MaxEaOld = Ea2
   End If

Loop

If IsBadFilePair = True Then Exit Sub

   AvgEa = (MaxEa + MinEa) / 2
   AvgKref = (Kref1 + Kref2) / 2

Open OutFileName For Append As #2
Write #2, FileName1, FileName2, AvgEa, AvgKref

Close #2

End Sub
```

```
Sub Print2Graph1 ()

Dim i As Integer

   OutEPM1.gphOut1.NumSets = 1
   OutEPM1.gphOut1.NumPoints = Nr
   OutEPM1.gphOut1.AutoInc = 1
   For i = 1 To Nr
      OutEPM1.gphOut1.GraphData = T(1, i)
   Next i
   OutEPM1.gphOut1.DrawMode = 2

End Sub

Sub Print2Graph2 ()
Dim i As Integer

   OutEPM1.gphOut2.NumSets = 2
   OutEPM1.gphOut2.NumPoints = GraphIndex
   OutEPM1.gphOut2.AutoInc = 0

   For i = 1 To 2
   OutEPM1.gphOut2.ThisSet = i
      For j = 1 To GraphIndex
      OutEPM1.gphOut2.ThisPoint = j
         OutEPM1.gphOut2.GraphData = GraphNode(i, j)
      Next j
   Next i

   OutEPM1.gphOut2.DrawMode = 2

End Sub

Private Sub Print2Grid ()

Dim i As Integer, j As Integer

   OutEPM1.txttNow.Text = Format(tNow / 60, "###.##")

End Sub

Function RootBisection (FileIndex As Integer, Gs() As Double, x1 As Double, x2 As
Double, xacc As Double) As Double
```

```
Dim fmid As Double, f As Double, dx As Double, xmid As Double
Dim bound As Double
ReDim Preserve Gs(Nr - 1, Ny - 1)

fmid = Kref_Gs_Func(FileIndex, Gs(), x2)
f = Kref_Gs_Func(FileIndex, Gs(), x1)

If f * fmid >= 0 Then
   IsBadFilePair = True
   Exit Function
ElseIf f < 0 Then
   bound = x1
   dx = x2 - x1
Else
   bound = x2
   dx = x1 - x2
End If

Do
   dx = dx * .5
   xmid = bound + dx
   fmid = Kref_Gs_Func(FileIndex, Gs(), xmid)
   If fmid <= 0 Then bound = xmid
   If Abs(dx) < xacc Or fmid = 0 Then Exit Do
Loop

RootBisection = bound

End Function

Private Sub SetFCylDimensions ()

ReDim rA(Nr, Ny) As Double, yA(Nr) As Double, dVol(Nr, Ny) As Double
ReDim r(Nr)  As Double, y(Ny) As Double
ReDim dr(Nr - 1) As Double, dy(Ny - 1) As Double
ReDim rALocation(Nr) As Double, yALocation(Ny) As Double

Dim yAsum As Double
Dim Pi As Double
Dim CheckVol

   Pi = 4 * Atn(1)

'  set internodal distances in radial direction
```

```
   dr(Nr - 1) = Radius / (2 * Nr - 3):  ' Smallest Common 'dr' Unit
   For i = 1 To Nr - 2
      dr(i) = 2 * dr(Nr - 1)
   Next i

'  set internodal dist's in height direction
   dy(Ny - 1) = Height / (2 * Ny - 3): ' Smallest common 'dy' Unit
   For i = 1 To Ny - 2
      dy(i) = 2 * dy(Ny - 1)
   Next i

'  set radial node locations
   r(1) = 0
   For i = 2 To Nr
      r(i) = r(i - 1) + dr(i - 1)
   Next i

'  set volume element bdy locations-(log mean radius)
   rALocation(1) = r(2) / 2
   For i = 2 To Nr - 1
      rALocation(i) = (r(i + 1) - r(i)) / (Log(r(i + 1) / r(i)))
   Next i
   rALocation(Nr) = Radius

'  set height node locations
   y(1) = 0
   For i = 2 To Ny
      y(i) = y(i - 1) + dy(i - 1)
   Next i

'  set locations for tops and bottoms of elements
   yALocation(1) = y(2) / 2
   For j = 2 To Ny - 1
      yALocation(j) = yALocation(j - 1) + dy(j - 1)
   Next j
   yALocation(Ny) = Height

'  Vertical areas - Radial heat flow
   For j = 1 To Ny
      For i = 1 To Nr
         rA(i, j) = 2 * Pi * rALocation(i) * (yALocation(j) - yALocation(j - 1))
      Next i
   Next j
```

```
' Horizontal Areas - tops of elements - vertical heat flow
  yA(1) = Pi * rALocation(1) * rALocation(1)
  yAsum = yA(1)
  For i = 2 To Nr - 2
     yA(i) = Pi * rALocation(i) * rALocation(i) - yAsum
     yAsum = yAsum + yA(i)
  Next i
  yA(Nr - 1) = Pi * Radius * Radius - yAsum

' Volumes
  CheckVol = 0
  For j = 1 To Ny
     For i = 1 To Nr
          dVol(i, j) = yA(i) * (yALocation(j) - yALocation(j - 1))
          CheckVol = CheckVol + dVol(i, j)
     Next i
  Next j
  FCylVol = CheckVol

End Sub

Private Sub SetFCylInitialTemp ()

ReDim T(Nr, Ny), Told(Nr, Ny)
ReDim Gs1(Nr - 1, Ny - 1), Gs2(Nr - 1, Ny - 1)
ReDim ConcRatio1(Nr - 1, Ny - 1), ConcRatio2(Nr - 1, Ny - 1)
'ReDim SimSurvConc(Nr - 1, Ny - 1)

' initialize temperatures
  For j = 1 To Ny - 1
     For i = 1 To Nr
        Told(i, j) = T0
     Next i
  Next j

  For i = 1 To Nr - 1
     Told(i, Ny) = T0
  Next i

'initialize G values and number of bugs per finite volume.
  For j = 1 To Ny - 1
     For i = 1 To Nr - 1
        Gs1(i, j) = 0
        Gs2(i, j) = 0
```

```
            ConcRatio1(i, j) = 0
            ConcRatio2(i, j) = 0
        Next i
    Next j

End Sub

Sub SimSurv ()
Dim Z As Double, kRate As Double, rxnTemp As Double

'   calc using 55% of temp change
    If T(i, j) > Told(i, j) Then
        rxnTemp = Told(i, j) + .55 * (T(i, j) - Told(i, j))
    Else
        rxnTemp = T(i, j) + .55 * (Told(i, j) - T(i, j))
    End If

    Z = ((121.1 + 273.15) - (rxnTemp + 273.15)) / ((121.1 + 273.15) * (rxnTemp +
273.15))
    kRate = K121Sim / 60 * Exp(-EaSim / Rg * Z)

    If SimSurvConc(i, j) > .001 Then
        SimSurvConc(i, j) = SimSurvConc(i, j) * Exp(-kRate * TimeStep)
    Else
        SimSurvConc(i, j) = 0
    End If

End Sub
```

# APPENDIX D
## MICROBIAL ENUMERATION: 0.0 kGy

| Process | Can ID | Dilution | Volume | Colonies per Volume | | | | Concentration |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | (CFU/ml) |
| | | (10^x) | (uL) | | | | | (C/Co) |
| 0 | 46 | -3 | 50 | 40 | 42 | 32 | 40 | |
| | | | | 40 | 47 | 41 | 40 | |
| | 20 | | | 48 | 53 | 43 | 50 | 9.04E+05 |
| | | | | 56 | 48 | 50 | 53 | 1.000 |
| 1 | 2 | -3 | 50 | 32 | 30 | 38 | 31 | |
| | | | | 32 | 28 | 28 | 30 | |
| | 5 | | | 34 | 31 | 32 | 34 | 6.59E+05 |
| | | | | 42 | 36 | 36 | 33 | 0.729 |
| 2 | 25 | -3 | 50 | 28 | 28 | 29 | 29 | |
| | | | | 39 | 27 | 35 | 22 | |
| | 35 | | | 37 | 34 | 31 | 27 | 6.33E+05 |
| | | | | 35 | 28 | 35 | 42 | 0.700 |
| 3 | 31 | -3 | 50 | 10 | 20 | 24 | 18 | |
| | | | | 17 | 16 | 19 | 16 | |
| | 34 | | | 17 | 22 | 20 | 18 | 3.90E+05 |
| | | | | 24 | 23 | 26 | 22 | 0.432 |
| 4 | 39 | -3 | 50 | 12 | 10 | 16 | 11 | |
| | | | | 14 | 12 | 13 | 13 | |
| | 29 | | | 10 | 9 | 13 | 11 | 2.46E+05 |
| | | | | 15 | 12 | 15 | 11 | 0.272 |
| 5 | 47 | -3 | 50 | 33 | 40 | 40 | 33 | |
| | | | | 31 | 35 | 30 | 31 | |
| | 48 | | | 26 | 40 | 33 | 35 | 6.56E+05 |
| | | | | 28 | 31 | 32 | 27 | 0.726 |
| 6 | 57 | -2 | 50 | 29 | 24 | 33 | 36 | |
| | | | | 31 | 30 | 41 | 30 | |
| | 62 | | | 24 | 38 | 38 | 24 | 6.19E+04 |
| | | | | 32 | 26 | 30 | 29 | 0.068 |
| 7 | 28 | -2 | 50 | 22 | 15 | 17 | 15 | |
| | | | | 26 | 19 | 16 | 21 | |
| | 51 | | | 20 | 19 | 11 | 7 | 3.70E+04 |
| | | | | 29 | 24 | 23 | 12 | 0.041 |
| 8 | 4 | -1 | 50 | 31 | 36 | 28 | 25 | |
| | | | | 36 | 36 | 30 | 40 | |
| | 21 | | | 51 | 36 | 46 | 39 | 7.14E+03 |
| | | | | 27 | 41 | 30 | 39 | 0.008 |

| Process | Can ID | Dilution | Volume | Colonies per Volume | | | | Concentration |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | (CFU/ml) |
| | | (10^x) | (uL) | | | | | (C/Co) |
| 9 | 206 | -3 | 50 | 32 | 29 | 33 | 26 | |
| | | | | 24 | 36 | 32 | 25 | 5.80E+05 |
| | | | | 27 | 26 | 33 | 25 | 0.642 |
| 10 | 214 | -3 | 50 | 21 | 18 | 16 | 17 | |
| | | | | 20 | 19 | 18 | 16 | 3.92E+05 |
| | | | | 19 | 19 | 29 | 23 | 0.433 |
| 11 | 216 | -3 | 50 | 9 | 13 | 12 | 4 | |
| | | | | 10 | 5 | 4 | 12 | 1.44E+05 |
| | | | | 3 | 5 | 5 | 4 | 0.159 |
| 12 | 203 | -3 | 50 | 20 | 18 | 33 | 28 | |
| | | | | 20 | 21 | 31 | 20 | 4.51E+05 |
| | | | | 19 | 25 | 18 | 18 | 0.499 |
| 13 | 213 | -3 | 50 | 11 | 9 | 9 | 18 | |
| | | | | 10 | 11 | 10 | 12 | 2.40E+05 |
| | | | | 17 | 14 | 11 | 12 | 0.265 |
| 14 | 217 | -2 | 50 | 50 | 47 | 36 | 48 | |
| | | | | 46 | 36 | 42 | 44 | 8.38E+04 |
| | | | | 35 | 39 | 40 | 40 | 0.093 |

# APPENDIX E
# MICROBIAL ENUMERATION: 1.0 kGy

| Process | Can ID | Dilution | Volume | Colonies per Volume | | | Concentration |
|---|---|---|---|---|---|---|---|
| | | | | | | | (CFU/ml) |
| | | (10^x) | (uL) | | | | (C/Co) |
| 0 | 162 | -3 | 50 | 49 | 43 | 42 | |
| | | | | 48 | 43 | 45 | |
| | | | | 44 | 45 | 41 | 8.82E+05 |
| | | | | 47 | 42 | 40 | 1.000 |
| 1 | 152 | -3 | 50 | 28 | 30 | 29 | |
| | | | | 27 | 20 | 20 | |
| | | | | 32 | 28 | 26 | 5.37E+05 |
| | | | | 25 | 32 | 25 | 0.609 |
| 2 | 149 | -3 | 50 | 26 | 21 | 28 | |
| | | | | 24 | 36 | 28 | |
| | | | | 23 | 18 | 23 | 5.10E+05 |
| | | | | 30 | 21 | 28 | 0.578 |
| 3 | 115 | -3 | 50 | 15 | 10 | 19 | |
| | | | | 8 | 17 | 19 | |
| | | | | 9 | 14 | 9 | 2.57E+05 |
| | | | | 7 | 14 | 13 | 0.291 |
| 4 | 159 | -2 | 50 | 91 | 61 | 52 | |
| | | | | 75 | 58 | 63 | |
| | | | | 71 | 60 | 74 | 1.33E+05 |
| | | | | 80 | 54 | 58 | 0.151 |
| 5 | 100 | -2 | 50 | 75 | 65 | 49 | |
| | | | | 56 | 65 | 65 | |
| | | | | 57 | 59 | 64 | 1.20E+05 |
| | | | | 57 | 56 | 50 | 0.136 |
| 6 | 143 | -1 | 50 | 50 | 45 | 68 | |
| | | | | 47 | 47 | 57 | |
| | | | | 44 | 50 | 58 | 1.07E+04 |
| | | | | 57 | 53 | 63 | 0.012 |
| 7 | 166 | -1 | 50 | 12 | 17 | 11 | |
| | | | | 13 | 10 | 16 | |
| | | | | 23 | 14 | 13 | 2.93E+03 |
| | | | | 19 | 13 | 15 | 0.003 |
| 8 | 156 | -1 | 50 | 1 | 1 | 2 | |
| | | | | 1 | 2 | 3 | |
| | | | | 3 | 4 | 0 | 3.17E+02 |
| | | | | 2 | 0 | 0 | 0.0004 |

| Process | Can ID | Dilution | Volume | Colonies per Volume | | | Concentration |
|---|---|---|---|---|---|---|---|
| | | | | | | | (CFU/ml) |
| | | (10^x) | (uL) | | | | (C/Co) |
| 11 | 167 | -3 | 50 | 14 | 21 | 23 | |
| | | | | 20 | 31 | 17 | |
| | | | | 29 | 26 | 19 | 4.43E+05 |
| | | | | 18 | 24 | 24 | 0.503 |
| 12 | 136 | -3 | 50 | 20 | 20 | 17 | |
| | | | | 16 | 18 | 17 | |
| | | | | 12 | 21 | 17 | 3.32E+05 |
| | | | | 9 | 24 | 8 | 0.376 |
| 13 | 126 | -3 | 50 | 14 | 21 | 13 | |
| | | | | 14 | 13 | 8 | |
| | | | | 19 | 19 | 9 | 2.82E+05 |
| | | | | 12 | 18 | 9 | 0.319 |
| 14 | 116 | -2 | 50 | 57 | 70 | 56 | |
| | | | | 44 | 68 | 71 | |
| | | | | 54 | 56 | 64 | 1.19E+05 |
| | | | | 67 | 54 | 54 | 0.135 |
| 16 | 146 | -3 | 50 | 14 | 18 | 25 | |
| | | | | 20 | 22 | 26 | |
| | | | | 18 | 17 | 16 | 3.75E+05 |
| | | | | 11 | 15 | 23 | 0.425 |
| 17 | 154 | -3 | 50 | 7 | 13 | 10 | |
| | | | | 10 | 9 | 18 | |
| | | | | 6 | 10 | 20 | 2.22E+05 |
| | | | | 11 | 12 | 7 | 0.251 |
| 18 | 124 | -3 | 50 | 15 | 10 | 13 | |
| | | | | 21 | 17 | 16 | |
| | | | | 12 | 10 | 18 | 2.83E+05 |
| | | | | 12 | 16 | 10 | 0.321 |
| 20 | 137 | -2 | 50 | 54 | 51 | 53 | |
| | | | | 54 | 46 | 52 | |
| | | | | 73 | 58 | 39 | 1.03E+05 |
| | | | | 48 | 47 | 42 | 0.117 |
| 21 | 160 | -2 | 50 | 36 | 37 | 39 | |
| | | | | 32 | 20 | 27 | |
| | | | | 27 | 21 | 38 | 6.60E+04 |
| | | | | 37 | 41 | 41 | 0.075 |
| 22 | 102 | -1 | 50 | 51 | 50 | 51 | |
| | | | | 64 | 50 | 62 | |
| | | | | 56 | 59 | 57 | 1.09E+04 |
| | | | | 44 | 53 | 58 | 0.012 |

## APPENDIX F
## MICROBIAL ENUMERATION: 3.0 kGy

| Process | Can ID | Dilution | Volume | Colonies per Volume | | | Concentration |
|---|---|---|---|---|---|---|---|
| | | | | | | | (CFU/ml) |
| | | (10^x) | (uL) | | | | (C/Co) |
| 0 | 117 | -3 | 50 | 33 | 38 | 47 | |
| | | | | 34 | 39 | 44 | |
| | | | | 40 | 44 | 40 | 7.73E+05 |
| | | | | 22 | 44 | 39 | 1.000 |
| 1 | 134 | -3 | 50 | 25 | 25 | 22 | |
| | | | | 20 | 22 | 21 | |
| | | | | 15 | 14 | 18 | 4.17E+05 |
| | | | | 32 | 19 | 17 | 0.539 |
| 2 | 112 | -3 | 50 | 22 | 28 | 22 | |
| | | | | 17 | 25 | 18 | |
| | | | | 14 | 15 | 20 | 4.10E+05 |
| | | | | 22 | 21 | 22 | 0.530 |
| 3 | 131 | -2 | 50 | 83 | 102 | 110 | |
| | | | | 87 | 101 | 87 | |
| | | | | 75 | 79 | 98 | 1.78E+05 |
| | | | | 80 | 84 | 79 | 0.230 |
| 4 | 163 | -2 | 50 | 18 | 28 | 24 | |
| | | | | 27 | 27 | 20 | |
| | | | | 30 | 22 | 29 | 4.95E+04 |
| | | | | 26 | 22 | 24 | 0.064 |
| 5 | 103 | -2 | 50 | 18 | 18 | 23 | |
| | | | | 27 | 14 | 23 | |
| | | | | 24 | 19 | 18 | 4.43E+04 |
| | | | | 23 | 28 | 31 | 0.057 |
| 6 | 107 | -1 | 50 | 9 | 9 | 13 | |
| | | | | 18 | 15 | 6 | |
| | | | | 11 | 6 | 11 | 2.37E+03 |
| | | | | 17 | 10 | 17 | 0.003 |

| Process | Can ID | Dilution | Volume | Colonies per Volume | | | Concentration |
|---|---|---|---|---|---|---|---|
| | | | | | | | (CFU/ml) |
| | | (10^x) | (uL) | | | | (C/Co) |
| 9 | 123 | -3 | 50 | 21 | 31 | 19 | |
| | | | | 24 | 21 | 19 | |
| | | | | 22 | 21 | 19 | 4.42E+05 |
| | | | | 23 | 22 | 23 | 0.571 |
| 10 | 106 | -3 | 50 | 24 | 19 | 21 | |
| | | | | 20 | 26 | 26 | |
| | | | | 15 | 16 | 23 | 4.07E+05 |
| | | | | 13 | 22 | 19 | 0.526 |
| 11 | 121 | -3 | 50 | 16 | 13 | 14 | |
| | | | | 26 | 12 | 16 | |
| | | | | 20 | 13 | 27 | 3.38E+05 |
| | | | | 16 | 12 | 18 | 0.438 |
| 12 | 110 | -3 | 50 | 11 | 11 | 16 | |
| | | | | 7 | 13 | 22 | |
| | | | | 7 | 9 | 13 | 2.28E+05 |
| | | | | 6 | 11 | 11 | 0.295 |
| 13 | 158 | -2 | 50 | 78 | 75 | 75 | |
| | | | | 79 | 85 | 75 | |
| | | | | 99 | 86 | 78 | 1.62E+05 |
| | | | | 97 | 74 | 73 | 0.210 |
| 14 | 165 | -2 | 50 | 31 | 39 | 38 | |
| | | | | 34 | 38 | 35 | |
| | | | | 40 | 37 | 30 | 6.85E+04 |
| | | | | 22 | 33 | 34 | 0.089 |

| Process | Can ID | Dilution | Volume | Colonies per Volume | | | Concentration |
|---|---|---|---|---|---|---|---|
| | | | | | | | (CFU/ml) |
| | | (10^x) | (uL) | | | | (C/Co) |
| 15 | 151 | -3 | 50 | 37 | 22 | 13 | |
| | | | | 11 | 18 | 12 | |
| | | | | 15 | 20 | 13 | 3.50E+05 |
| | | | | 21 | 18 | 10 | 0.453 |
| 16 | 104 | -3 | 50 | 16 | 14 | 18 | |
| | | | | 12 | 14 | 11 | |
| | | | | 11 | 10 | 13 | 2.58E+05 |
| | | | | 7 | 13 | 16 | 0.334 |
| 17 | 118 | -3 | 50 | 9 | 9 | 11 | |
| | | | | 7 | 6 | 10 | |
| | | | | 11 | 6 | 13 | 1.97E+05 |
| | | | | 13 | 13 | 10 | 0.254 |
| 18 | 128 | -2 | 50 | 67 | 72 | 77 | |
| | | | | 68 | 67 | 70 | |
| | | | | 70 | 92 | 68 | 1.49E+05 |
| | | | | 84 | 85 | 74 | 0.193 |
| 19 | 147 | -2 | 50 | 54 | 44 | 40 | |
| | | | | 45 | 50 | 42 | |
| | | | | 45 | 44 | 59 | 9.40E+04 |
| | | | | 50 | 43 | 48 | 0.122 |
| 20 | 114 | -2 | 50 | 37 | 27 | 42 | |
| | | | | 26 | 27 | 41 | |
| | | | | 30 | 30 | 26 | 6.57E+04 |
| | | | | 31 | 42 | 35 | 0.085 |
| 21 | 130 | -2 | 50 | 15 | 13 | 18 | |
| | | | | 17 | 8 | 20 | |
| | | | | 17 | 16 | 11 | 2.80E+04 |
| | | | | 14 | 13 | 6 | 0.036 |
| 22 | 153 | -1 | 50 | 14 | 11 | 5 | |
| | | | | 11 | 8 | 14 | |
| | | | | 15 | 8 | 15 | 2.08E+03 |
| | | | | 5 | 9 | 10 | 0.003 |

# APPENDIX G
# PROCESS PAIR DATA: 0.0 kGy

| Process Pair | Process | | Process | | Ea | k121 |
|---|---|---|---|---|---|---|
| Index | # | (C/Co) | # | (C/Co) | (J/mol) | (1/min) |
| 1 | 1 | 0.725 | 9 | 0.642 | 198,900 | 0.138 |
| 2 | 1 | 0.725 | 10 | 0.434 | 241,772 | 0.256 |
| 3 | 1 | 0.725 | 11 | 0.159 | 276,506 | 0.421 |
| 4 | 1 | 0.725 | 12 | 0.499 | 222,852 | 0.195 |
| 5 | 1 | 0.725 | 13 | 0.265 | 237,293 | 0.240 |
| 6 | 1 | 0.725 | 14 | 0.093 | 255,632 | 0.312 |
| 7 | 8 | 0.008 | 11 | 0.159 | 198,280 | 0.234 |
| 8 | 8 | 0.008 | 12 | 0.499 | 154,468 | 0.130 |
| 9 | 8 | 0.008 | 13 | 0.265 | 179,467 | 0.182 |
| 10 | 8 | 0.008 | 14 | 0.093 | 205,742 | 0.259 |
| 11 | 2 | 0.700 | 10 | 0.434 | 346,349 | 0.591 |
| 12 | 2 | 0.700 | 11 | 0.159 | 371,003 | 0.836 |
| 13 | 2 | 0.700 | 12 | 0.499 | 294,807 | 0.285 |
| 14 | 2 | 0.700 | 13 | 0.265 | 303,346 | 0.322 |
| 15 | 2 | 0.700 | 14 | 0.093 | 316,304 | 0.386 |
| 16 | 3 | 0.431 | 9 | 0.642 | 290,312 | 0.313 |
| 17 | 3 | 0.431 | 10 | 0.434 | 322,619 | 0.490 |
| 18 | 3 | 0.431 | 11 | 0.159 | 350,471 | 0.721 |
| 19 | 3 | 0.431 | 12 | 0.499 | 277,277 | 0.261 |
| 20 | 3 | 0.431 | 13 | 0.265 | 287,424 | 0.300 |
| 21 | 3 | 0.431 | 14 | 0.093 | 302,017 | 0.368 |
| 22 | 4 | 0.272 | 10 | 0.434 | 283,962 | 0.360 |
| 23 | 4 | 0.272 | 11 | 0.159 | 317,546 | 0.568 |
| 24 | 4 | 0.272 | 12 | 0.499 | 249,393 | 0.225 |
| 25 | 4 | 0.272 | 13 | 0.265 | 262,451 | 0.269 |
| 26 | 4 | 0.272 | 14 | 0.093 | 279,811 | 0.341 |

| Process Pair | Process | | Process | | Ea | k121 |
|---|---|---|---|---|---|---|
| Index | # | (C/Co) | # | (C/Co) | (J/mol) | (1/min) |
| 27 | 5 | 0.072 | 9 | 0.642 | 152,401 | 0.089 |
| 28 | 5 | 0.072 | 10 | 0.434 | 212,025 | 0.200 |
| 29 | 5 | 0.072 | 11 | 0.159 | 254,956 | 0.359 |
| 30 | 5 | 0.072 | 12 | 0.499 | 201,055 | 0.172 |
| 31 | 5 | 0.072 | 13 | 0.265 | 219,184 | 0.221 |
| 32 | 5 | 0.072 | 14 | 0.093 | 240,686 | 0.296 |
| 33 | 6 | 0.068 | 9 | 0.642 | 141,918 | 0.080 |
| 34 | 6 | 0.068 | 10 | 0.434 | 203,143 | 0.186 |
| 35 | 6 | 0.068 | 11 | 0.159 | 246,710 | 0.338 |
| 36 | 6 | 0.068 | 12 | 0.499 | 195,218 | 0.167 |
| 37 | 6 | 0.068 | 13 | 0.265 | 213,835 | 0.215 |
| 38 | 6 | 0.068 | 14 | 0.093 | 235,660 | 0.290 |
| 39 | 7 | 0.041 | 9 | 0.642 | 143,983 | 0.082 |
| 40 | 7 | 0.041 | 10 | 0.434 | 209,489 | 0.196 |
| 41 | 7 | 0.041 | 11 | 0.159 | 254,957 | 0.359 |
| 42 | 7 | 0.041 | 12 | 0.499 | 198,965 | 0.170 |
| 43 | 7 | 0.041 | 13 | 0.265 | 217,967 | 0.219 |
| 44 | 7 | 0.041 | 14 | 0.093 | 240,251 | 0.295 |
| 45 | 9 | 0.642 | 12 | 0.499 | 258,513 | 0.236 |
| 46 | 9 | 0.642 | 13 | 0.265 | 283,987 | 0.296 |
| 47 | 9 | 0.642 | 14 | 0.093 | 313,726 | 0.383 |
| 48 | 10 | 0.434 | 13 | 0.265 | 229,657 | 0.232 |
| 49 | 10 | 0.434 | 14 | 0.093 | 274,689 | 0.334 |
| 50 | 11 | 0.159 | 13 | 0.265 | 138,929 | 0.147 |
| 51 | 11 | 0.159 | 14 | 0.093 | 217,313 | 0.271 |

## APPENDIX H
## PROCESS PAIR DATA: 1.0 kGy

| Process Pair | Process | | Process | | Ea | k121 |
|---|---|---|---|---|---|---|
| Index | # | (C/Co) | # | (C/Co) | (J/mol) | (1/min) |
| 1 | 7 | 0.003 | 11 | 0.502 | 88,052 | 0.088 |
| 2 | 7 | 0.003 | 12 | 0.376 | 125,436 | 0.149 |
| 3 | 7 | 0.003 | 13 | 0.320 | 91,297 | 0.092 |
| 4 | 7 | 0.003 | 14 | 0.135 | 123,588 | 0.145 |
| 5 | 7 | 0.003 | 22 | 0.012 | 187,087 | 0.349 |
| 6 | 7 | 0.003 | 16 | 0.425 | 166,003 | 0.261 |
| 7 | 7 | 0.003 | 17 | 0.252 | 179,231 | 0.313 |
| 8 | 7 | 0.003 | 18 | 0.320 | 121,904 | 0.142 |
| 9 | 7 | 0.003 | 20 | 0.117 | 154,883 | 0.224 |
| 10 | 7 | 0.003 | 21 | 0.075 | 152,068 | 0.215 |
| 11 | 8 | 0.0004 | 22 | 0.012 | 176,665 | 0.336 |
| 12 | 8 | 0.0004 | 16 | 0.425 | 145,246 | 0.220 |
| 13 | 8 | 0.0004 | 17 | 0.252 | 162,745 | 0.279 |
| 14 | 8 | 0.0004 | 18 | 0.320 | 101,991 | 0.123 |
| 15 | 8 | 0.0004 | 20 | 0.117 | 141,053 | 0.208 |
| 16 | 8 | 0.0004 | 21 | 0.075 | 139,379 | 0.204 |
| 17 | 1 | 0.609 | 11 | 0.502 | 144,115 | 0.167 |
| 18 | 1 | 0.609 | 12 | 0.376 | 157,479 | 0.205 |
| 19 | 1 | 0.609 | 13 | 0.320 | 129,817 | 0.133 |
| 20 | 1 | 0.609 | 14 | 0.135 | 147,869 | 0.177 |
| 21 | 1 | 0.609 | 16 | 0.425 | 180,699 | 0.293 |
| 22 | 1 | 0.609 | 17 | 0.252 | 189,953 | 0.338 |
| 23 | 1 | 0.609 | 18 | 0.320 | 142,611 | 0.163 |
| 24 | 1 | 0.609 | 20 | 0.117 | 167,154 | 0.238 |
| 25 | 1 | 0.609 | 21 | 0.075 | 163,841 | 0.226 |

| Process Pair | Process | | Process | | Ea | k121 |
|---|---|---|---|---|---|---|
| Index | # | (C/Co) | # | (C/Co) | (J/mol) | (1/min) |
| 26 | 2 | 0.578 | 11 | 0.502 | 267,925 | 0.574 |
| 27 | 2 | 0.578 | 12 | 0.376 | 267,858 | 0.573 |
| 28 | 2 | 0.578 | 13 | 0.320 | 219,064 | 0.286 |
| 29 | 2 | 0.578 | 14 | 0.135 | 229,087 | 0.330 |
| 30 | 2 | 0.578 | 16 | 0.425 | 264,544 | 0.547 |
| 31 | 2 | 0.578 | 17 | 0.252 | 266,438 | 0.562 |
| 32 | 2 | 0.578 | 18 | 0.320 | 206,761 | 0.240 |
| 33 | 2 | 0.578 | 20 | 0.117 | 225,415 | 0.313 |
| 34 | 2 | 0.578 | 21 | 0.075 | 217,253 | 0.279 |
| 35 | 2 | 0.578 | 22 | 0.012 | 247,244 | 0.428 |
| 36 | 3 | 0.291 | 12 | 0.376 | 235,686 | 0.428 |
| 37 | 3 | 0.291 | 13 | 0.320 | 186,521 | 0.218 |
| 38 | 3 | 0.291 | 14 | 0.135 | 201,513 | 0.268 |
| 39 | 3 | 0.291 | 16 | 0.425 | 241,200 | 0.462 |
| 40 | 3 | 0.291 | 17 | 0.252 | 245,574 | 0.490 |
| 41 | 3 | 0.291 | 18 | 0.320 | 183,663 | 0.210 |
| 42 | 3 | 0.291 | 20 | 0.117 | 206,616 | 0.288 |
| 43 | 3 | 0.291 | 21 | 0.075 | 199,496 | 0.261 |
| 44 | 3 | 0.291 | 22 | 0.012 | 231,981 | 0.407 |
| 45 | 4 | 0.151 | 12 | 0.376 | 238,694 | 0.440 |
| 46 | 4 | 0.151 | 13 | 0.320 | 188,935 | 0.223 |
| 47 | 4 | 0.151 | 14 | 0.135 | 203,688 | 0.273 |
| 48 | 4 | 0.151 | 16 | 0.425 | 243,340 | 0.469 |
| 49 | 4 | 0.151 | 17 | 0.252 | 247,491 | 0.497 |
| 50 | 4 | 0.151 | 18 | 0.320 | 185,281 | 0.212 |
| 51 | 4 | 0.151 | 20 | 0.117 | 208,090 | 0.290 |
| 52 | 4 | 0.151 | 21 | 0.075 | 200,833 | 0.262 |
| 53 | 4 | 0.151 | 22 | 0.012 | 233,294 | 0.409 |

| Process Pair | Process | | Process | | Ea | k121 |
|---|---|---|---|---|---|---|
| Index | # | (C/Co) | # | (C/Co) | (J/mol) | (1/min) |
| 54 | 5 | 0.136 | 12 | 0.376 | 292,819 | 0.717 |
| 55 | 5 | 0.136 | 13 | 0.320 | 233,139 | 0.321 |
| 56 | 5 | 0.136 | 14 | 0.135 | 242,827 | 0.366 |
| 57 | 5 | 0.136 | 16 | 0.425 | 282,021 | 0.620 |
| 58 | 5 | 0.136 | 17 | 0.252 | 282,387 | 0.623 |
| 59 | 5 | 0.136 | 18 | 0.320 | 214,951 | 0.251 |
| 60 | 5 | 0.136 | 20 | 0.117 | 234,239 | 0.326 |
| 61 | 5 | 0.136 | 21 | 0.075 | 224,669 | 0.286 |
| 62 | 5 | 0.136 | 22 | 0.012 | 256,643 | 0.441 |
| 63 | 6 | 0.012 | 11 | 0.502 | 147,650 | 0.173 |
| 64 | 6 | 0.012 | 12 | 0.376 | 167,887 | 0.227 |
| 65 | 6 | 0.012 | 13 | 0.320 | 124,670 | 0.127 |
| 66 | 6 | 0.012 | 14 | 0.135 | 151,174 | 0.181 |
| 67 | 6 | 0.012 | 16 | 0.425 | 195,627 | 0.329 |
| 68 | 6 | 0.012 | 17 | 0.252 | 205,557 | 0.376 |
| 69 | 6 | 0.012 | 18 | 0.320 | 143,556 | 0.164 |
| 70 | 6 | 0.012 | 20 | 0.117 | 174,015 | 0.246 |
| 71 | 6 | 0.012 | 21 | 0.075 | 169,384 | 0.231 |
| 72 | 6 | 0.012 | 22 | 0.012 | 204,779 | 0.372 |
| 73 | 11 | 0.502 | 16 | 0.425 | 257,926 | 0.522 |
| 74 | 11 | 0.502 | 17 | 0.252 | 264,184 | 0.554 |
| 75 | 11 | 0.502 | 18 | 0.320 | 140,839 | 0.161 |
| 76 | 11 | 0.502 | 20 | 0.117 | 188,781 | 0.265 |
| 77 | 11 | 0.502 | 21 | 0.075 | 179,780 | 0.241 |
| 78 | 11 | 0.502 | 22 | 0.012 | 232,492 | 0.408 |

| Process Pair | Process | | Process | | Ea | k121 |
|---|---|---|---|---|---|---|
| Index | # | (C/Co) | # | (C/Co) | (J/mol) | (1/min) |
| 79 | 12 | 0.376 | 22 | 0.012 | 229,133 | 0.403 |
| 80 | 12 | 0.376 | 17 | 0.252 | 263,491 | 0.551 |
| 81 | 12 | 0.376 | 18 | 0.320 | 118,202 | 0.138 |
| 82 | 12 | 0.376 | 20 | 0.117 | 178,928 | 0.252 |
| 83 | 12 | 0.376 | 21 | 0.075 | 170,396 | 0.232 |
| 84 | 12 | 0.376 | 22 | 0.012 | 229,133 | 0.403 |
| 85 | 13 | 0.320 | 22 | 0.012 | 282,079 | 0.476 |
| 86 | 13 | 0.320 | 18 | 0.320 | 177,641 | 0.202 |
| 87 | 13 | 0.320 | 20 | 0.117 | 235,807 | 0.328 |
| 88 | 13 | 0.320 | 21 | 0.075 | 214,838 | 0.276 |
| 89 | 13 | 0.320 | 22 | 0.012 | 282,079 | 0.476 |
| 90 | 14 | 0.135 | 22 | 0.012 | 276,111 | 0.468 |
| 91 | 14 | 0.135 | 18 | 0.320 | 114,832 | 0.135 |
| 92 | 14 | 0.135 | 21 | 0.075 | 196,241 | 0.258 |
| 93 | 14 | 0.135 | 22 | 0.012 | 276,111 | 0.468 |

# APPENDIX I
## PROCESS PAIR DATA: 3.0 kGy

| Process Pair | Process | | Process | | Ea | k121 |
|---|---|---|---|---|---|---|
| Index | # | (C/Co) | # | (C/Co) | (J/mol) | (1/min) |
| 1 | 1 | 0.539 | 10 | 0.526 | 274,356 | 1.501 |
| 2 | 1 | 0.539 | 11 | 0.473 | 138,552 | 0.190 |
| 3 | 1 | 0.539 | 12 | 0.295 | 160,101 | 0.267 |
| 4 | 1 | 0.539 | 13 | 0.210 | 145,592 | 0.213 |
| 5 | 1 | 0.539 | 14 | 0.089 | 144,506 | 0.209 |
| 6 | 1 | 0.539 | 16 | 0.334 | 199,021 | 0.484 |
| 7 | 1 | 0.539 | 17 | 0.255 | 161,901 | 0.274 |
| 8 | 1 | 0.539 | 18 | 0.193 | 166,398 | 0.294 |
| 9 | 1 | 0.539 | 19 | 0.122 | 165,480 | 0.290 |
| 10 | 1 | 0.539 | 20 | 0.085 | 161,168 | 0.271 |
| 11 | 1 | 0.539 | 21 | 0.036 | 169,278 | 0.307 |
| 12 | 2 | 0.530 | 11 | 0.473 | 281,814 | 0.810 |
| 13 | 2 | 0.530 | 12 | 0.295 | 288,035 | 0.885 |
| 14 | 2 | 0.530 | 13 | 0.210 | 250,695 | 0.521 |
| 15 | 2 | 0.530 | 14 | 0.089 | 236,722 | 0.427 |
| 16 | 2 | 0.530 | 16 | 0.334 | 312,006 | 1.241 |
| 17 | 2 | 0.530 | 17 | 0.255 | 245,540 | 0.484 |
| 18 | 2 | 0.530 | 18 | 0.193 | 247,336 | 0.496 |
| 19 | 2 | 0.530 | 19 | 0.122 | 237,140 | 0.429 |
| 20 | 2 | 0.530 | 20 | 0.085 | 227,800 | 0.376 |
| 21 | 2 | 0.530 | 21 | 0.036 | 231,446 | 0.396 |
| 22 | 3 | 0.230 | 11 | 0.473 | 238,963 | 0.534 |
| 23 | 3 | 0.230 | 12 | 0.295 | 251,630 | 0.635 |
| 24 | 3 | 0.230 | 13 | 0.210 | 216,353 | 0.392 |
| 25 | 3 | 0.230 | 14 | 0.089 | 205,222 | 0.336 |
| 26 | 3 | 0.230 | 16 | 0.334 | 284,027 | 0.988 |
| 27 | 3 | 0.230 | 17 | 0.255 | 218,938 | 0.406 |
| 28 | 3 | 0.230 | 18 | 0.193 | 221,971 | 0.423 |
| 29 | 3 | 0.230 | 19 | 0.122 | 214,109 | 0.380 |
| 30 | 3 | 0.230 | 20 | 0.085 | 205,704 | 0.338 |
| 31 | 3 | 0.230 | 21 | 0.036 | 211,375 | 0.366 |

| Process Pair | Process | | Process | | Ea | k121 |
|---|---|---|---|---|---|---|
| Index | # | (C/Co) | # | (C/Co) | (J/mol) | (1/min) |
| 32 | 4 | 0.064 | 11 | 0.473 | 188,848 | 0.323 |
| 33 | 4 | 0.064 | 12 | 0.295 | 209,406 | 0.429 |
| 34 | 4 | 0.064 | 13 | 0.210 | 181,045 | 0.290 |
| 35 | 4 | 0.064 | 14 | 0.089 | 174,458 | 0.265 |
| 36 | 4 | 0.064 | 16 | 0.334 | 249,433 | 0.742 |
| 37 | 4 | 0.064 | 17 | 0.255 | 192,339 | 0.339 |
| 38 | 4 | 0.064 | 18 | 0.193 | 196,510 | 0.359 |
| 39 | 4 | 0.064 | 19 | 0.122 | 191,698 | 0.336 |
| 40 | 4 | 0.064 | 20 | 0.085 | 184,772 | 0.306 |
| 41 | 4 | 0.064 | 21 | 0.036 | 192,221 | 0.339 |
| 42 | 5 | 0.057 | 11 | 0.473 | 257,835 | 0.642 |
| 43 | 5 | 0.057 | 12 | 0.295 | 268,358 | 0.740 |
| 44 | 5 | 0.057 | 13 | 0.210 | 228,861 | 0.435 |
| 45 | 5 | 0.057 | 14 | 0.089 | 215,700 | 0.364 |
| 46 | 5 | 0.057 | 16 | 0.334 | 298,928 | 1.116 |
| 47 | 5 | 0.057 | 17 | 0.255 | 228,438 | 0.432 |
| 48 | 5 | 0.057 | 18 | 0.193 | 231,158 | 0.448 |
| 49 | 5 | 0.057 | 19 | 0.122 | 221,950 | 0.396 |
| 50 | 5 | 0.057 | 20 | 0.085 | 212,783 | 0.350 |
| 51 | 5 | 0.057 | 21 | 0.036 | 217,996 | 0.376 |
| 52 | 6 | 0.003 | 12 | 0.295 | 158,423 | 0.262 |
| 53 | 6 | 0.003 | 13 | 0.210 | 137,252 | 0.197 |
| 54 | 6 | 0.003 | 14 | 0.089 | 137,157 | 0.197 |
| 55 | 6 | 0.003 | 16 | 0.334 | 213,497 | 0.548 |
| 56 | 6 | 0.003 | 17 | 0.255 | 161,547 | 0.274 |
| 57 | 6 | 0.003 | 18 | 0.193 | 167,457 | 0.296 |
| 58 | 6 | 0.003 | 19 | 0.122 | 166,156 | 0.291 |
| 59 | 6 | 0.003 | 20 | 0.085 | 160,715 | 0.270 |
| 60 | 6 | 0.003 | 21 | 0.036 | 170,692 | 0.309 |

| Process Pair | Process | | Process | | Ea | k121 |
|---|---|---|---|---|---|---|
| Index | # | (C/Co) | # | (C/Co) | (J/mol) | (1/min) |
| 61 | 9 | 0.572 | 15 | 0.453 | 189,379 | 1.166 |
| 62 | 10 | 0.526 | 16 | 0.334 | 132,924 | 0.269 |
| 63 | 10 | 0.526 | 17 | 0.255 | 99,809 | 0.170 |
| 64 | 10 | 0.526 | 18 | 0.193 | 109,091 | 0.194 |
| 65 | 10 | 0.526 | 19 | 0.122 | 115,951 | 0.214 |
| 66 | 10 | 0.526 | 20 | 0.085 | 114,365 | 0.209 |
| 67 | 10 | 0.526 | 21 | 0.036 | 129,078 | 0.256 |
| 68 | 11 | 0.473 | 17 | 0.255 | 196,044 | 0.348 |
| 69 | 11 | 0.473 | 18 | 0.193 | 204,043 | 0.377 |
| 70 | 11 | 0.473 | 19 | 0.122 | 193,864 | 0.340 |
| 71 | 11 | 0.473 | 20 | 0.085 | 182,070 | 0.302 |
| 72 | 11 | 0.473 | 21 | 0.036 | 194,192 | 0.341 |
| 73 | 12 | 0.295 | 17 | 0.255 | 165,641 | 0.282 |
| 74 | 12 | 0.295 | 18 | 0.193 | 178,229 | 0.318 |
| 75 | 12 | 0.295 | 19 | 0.122 | 173,029 | 0.303 |
| 76 | 12 | 0.295 | 20 | 0.085 | 162,461 | 0.273 |
| 77 | 12 | 0.295 | 21 | 0.036 | 178,810 | 0.320 |
| 78 | 13 | 0.210 | 17 | 0.255 | 227,139 | 0.429 |
| 79 | 13 | 0.210 | 18 | 0.193 | 237,148 | 0.466 |
| 80 | 13 | 0.210 | 19 | 0.122 | 210,122 | 0.372 |
| 81 | 13 | 0.210 | 20 | 0.085 | 190,088 | 0.314 |
| 82 | 13 | 0.210 | 21 | 0.036 | 205,364 | 0.357 |
| 83 | 14 | 0.089 | 17 | 0.255 | 308,292 | 0.724 |
| 84 | 14 | 0.089 | 18 | 0.193 | 304,452 | 0.704 |
| 85 | 14 | 0.089 | 19 | 0.122 | 238,390 | 0.432 |
| 86 | 14 | 0.089 | 20 | 0.085 | 206,743 | 0.340 |
| 87 | 14 | 0.089 | 21 | 0.036 | 221,711 | 0.381 |

# REFERENCES

Agarwal, A.K. and Brisk, M.L. 1984. Sequential experimental design for precise parameter estimation. Ind. Eng. Chem. Process Des. Dev. 24:203-207.

Ahmed, N.M., Conner, D.E. and Huffman, D.L. 1995. Heat-resistance of *Escherichia coli* (O157:H7) in meat and poultry as affected by product composition. J. Food Sci. 60(3):606-610.

Alderton, G., Chen, J.K. and Ito, K.A. 1974. Effect of lysozyme on the recovery of heated *Clostridium botulinum* spores. Appl. Mircrobiol. 27(3):613-615.

Bellagha, S. and Chau, K.V. 1985. Heat and mass transfer during cooling of tomatoes individually and in bulk. Paper No. 85-6002. ASAE, St. Joseph, MI.

Berry, M.F., Singh, R.K. and Nelson, P.E. 1990. Kinetics of Methionine sulfonium salt destruction in a model particulate system. J. Food Sci. 55(2):502-505.

Biegler, L.T., Damiano, J.J. and Blau, G.E. 1986. Nonlinear parameter estimation: A case study comparison. AIChE J. 32(1):29-45.

Bruhn, C.M. and Noell, J.W. 1987. Consumer in-store response to irradiated papayas. Food Technol. 41(9):84-85.

Chapman, A.J. 1984. *Heat Transfer*, 4th ed. Macmillan Publishing Company, New York.

Chai, T.J. and Liang, K.T. 1992. Thermal resistance of spores from five type E *Clostridium botulinum* strains in eastern oyster homogenates. J. Food Protect. 55(1):18-22.

Chau, K.V. and Gaffney, J.J. 1990. A finite-difference model for heat and mass transfer in products with internal heat generation and transpiration. J. Food Sci. 55(2), 484-487.

Clausing, A.M. 1981. Numerical methods in heat transfer. Lectures in mechanical engineering, University of Illinois, Urbana, Illinois, USA.

Cohen, E. and Saguy, I. 1985. Statistical evaluation of Arrhenius model and its application in prediction of food quality losses. J. Food Proc. Preserv. 9:273-290.

Constantinou, C.P. 1994. The study of decomposition reactions through the exchange of thermal energy. J. Chem. Kinet. 26:1151-1166.

Deasy, P.B., Kuster, E. and Timoney, R.F. 1970. Resistance of *Bacillus subtilis* spores to inactivation by gamma irradiation and heating in the presence of bactericide. Appl. Microbiol. 20(3):461-464.

Diehl, J.F. 1990. *Safety of Irradiated Foods*. Marcel Dekker, Inc., New York.

Emborg, C. 1974. Inactivation of dried bacteria and bacterial spores by means of gamma irradiation at high temperatures. Appl. Microbiol. 27:830-833.

Espie, D.M. and Macchietto, S.M. 1988. Nonlinear transformations for parameter estimation. Ind. Eng. Chem. Res. 27:2175-2179.

Esty, J.R. and Meyer, K.F. 1922. The heat resistance of *B. botulinus* spores. J. Infect. Dis. 31:650.

Fernandez, P.S., Gomez, F.J., Ocio, M.J., Rodrigo, M., Sanchez, T. and Martinez, A. 1995. D values of Bacillus stearothermophilus spores as a function of pH and recovery medium acidulant. J. Food Protect. 58(6):628-632.

Fernandez, P.S., Ocio, M.J., Sanchez, T. and Martinez, A. 1994. Thermal resistance of *Bacillus stearothermophilus* spores heated in acidified mushroom extract. J. Food Protect. 57(1):37-41.

Fisher, D.A. and Pflug, I.J. 1977. Effect of combined heat and radiation on microbial destruction. Appl. Environ. Microbiol. 33(5):1170-1176.

Giese, J. 1994. Ultrapasteurized liquid whole eggs earn 1994 IFT Food Technology Industrial Achievement Award. Food Technol. 48(9):94-96.

Gombas, D.E. and Gomez, R.F. 1978. Sensitization of *Clostridium perfringes* spores to heat by gamma radiation. Appl. Environ. Microbiol. 36(3):403-407.

Gonzalez, I., Lopez, M., Mazas, M., Gonzalez, J. and Bernardo, A. 1995. The effect of recovery conditions on the apparent heat resistance of *Bacillus cereus* spores. J. Appl. Bacteriol. 78:548-554.

Gould, G.W. 1985. Modification of resistance and dormancy. In *Fundamental and Applied Aspects of Bacterial Spores*, G.F. Dring, D.F. Ellar and G.W. Gould (ed.), p. 371-382. Academic Press, Inc. New York.

Grecz, N., Walker, A.A., Anellis, A and Berkowitz, D. 1971. Effect of irradiation in the range of -196 to 95°C on the resistance of spores of *Clostridium botulinum* 33A in cooked beef. Can. J. Microbiol. 17(2):135-142.

Haberman, R. 1987. *Elementary Applied Partial Differential Equations*, 2nd ed. Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

Hayakawa, K.I., Schnell, P.G. and Kleyn, D.H. 1969. Estimating thermal death time characteristics of thermally vulnerable factors by programmed heating of sample solution or suspension. Food Technol. 23(8):104-109.

Juneja, V.K., Eblen, B.S., Marmer, B.S., Williams, A.C., Palumbo, S.A. and Miller, A.J. 1995. Thermal resistance of nonproteolytic type B and type E *Clostridium botulinum* spores in phosphate buffer and turkey slurry. J. Food Protect. 58(7):758-763.

Kempe, L.L. 1955. Combined effects of heat and radiation in food sterilization. Appl. Microbiol. 3:346-352.

Kempe, L.L., Graikoski, J.T. and Bonventre, P.F. 1957. Combined irradiation-heat processing of canned foods I. Appl. Microbiol. 5:292-295.

Kempe, L.L., Graikoski, J.T. and Bonventre, P.F. 1958. Combined irradiation-heat processing of canned foods II. Appl. Microbiol. 6:261-263.

Kempe, L.L., Graikoski, J.T. and Bonventre, P.F. 1959. Combined irradiation-heat processing of canned foods III. Appl. Microbiol. 7:131-2134.

Kempe, L.L., Graikoski, J.T. and Bonventre, P.F. 1960. Combined irradiation-heat processing of canned foods: Green pea inoculated with anaerobic bacterial spores. J. Biochem. Microbiol. Technol. Eng. 2(1):1-8.

Kim, I.W., Liebman, M.J. and Edgar, T.F. 1990. Robust error-in-variables estimation using nonlinear programming techniques. AIChE J. 36(7):985-993.

Labuza, T.P. 1979. A theoretical comparison of losses in foods under fluctuating temperature sequences. J. Food Sci. 44:1162-1168.

Lenius, P. 1995. Food irradiation: Has its time come? Prepared Foods (9):15-18.

Lenz, M.K. and Lund, D.B. 1980. Experimental procedures for determining destruction kinetics of food components. Food Technol. 34(2):51-55.

Licciardello, J.J. 1964. Effect of temperature on radiosensitivity of *Salmonella typhimurium*. J. Food Sci. 29:469-474.

Licciardello, J.J. and Nickerson, J.T.R. 1962. Effect of radiation environment on the thermal resistance of irradiated spores of *Clostridium sporogenes* PA 3679. J. Food Sci. 27:211-218.

Licciardello, J.J. and Nickerson, J.T.R. 1963. Effect of radiation environment on the thermal resistance of irradiated spores of *Bacillus subtilis*. Appl. Microbiol. 11:216-219.

Lienhard, J.H. 1987. *A Heat Transfer Textbook*, 2nd ed. Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

Lobo, L.S. and Lobo, M.S. 1991. Robust and efficient nonlinear regression of kinetic systems using a direct search method. Computers Chem. Engng. 15(2):141-144.

Lund, D.B. 1982. Influence of processing on nutrients in foods. J. Food Protect. 45, 367-373.

Lynch, D.J. and Potter, N.N. 1988. Effects of organic acids on thermal inactivation of *Bacillus stearothermophilus* and *Bacillus coagulans* spores in frankfurter emulsion slurry. J. Food Protect. 51(6):475-480.

Lynch, D.J. and Potter, N.N. 1989. Effects of acidification and processing variables on thermal inactivation of *Bacillus coagulans* spores in meat particulates. J. Food Protect. 52(5):320-328.

Maesmans, G., Hendrickx, M., DeCordt, S. and Tobback, P. 1995. Theoretical consideration of the general validity of the equivalent point method in thermal process evaluation. J. Food Eng. 24:225-248.

Mallidis, C.G. and Drizou, D. 1991. Effect of simultaneous application of heat and pressure on the survival of bacterial spores. J. Appl. Bacteriol. 71:285-288.

Marquardt, D.W. 1959. Solution of nonlinear chemical engineering models. Chem. Eng. Progress. 55(6):65-70.

Marquardt, D.W. 1963. An algorithm for least-squares estimation of nonlinear parameters. J. Soc. Indust. Appl. Math. 11(2):431-441.

Michener, H.D., Thompson, P.A. and Lewis, J.C. 1959. Search for substances which reduce the heat resistance of bacterial spores. Appl. Microbiol. 7:166-173.

Miltz, J. 1992. Food packaging. Chpt. 13. In *Handbook of Food Engineering*, ed. D.R. Heldman and D.B. Lund, p. 669. Marcel-Dekker, Inc., New York.

Nash, J.C. 1977. Minimizing a nonlinear sum of squares function on a small computer. J. Inst. Math. Applicat. 19:231-237.

Nasri, H., Simpson, R., Bouzas, J. and Torres, J.A. 1993. An unsteady-state method to determine kinetic parameters for heat inactivation of quality factors: Conduction-heated foods. J. Food Eng. 19:291-301.

Ozisik, M.N. 1985. *Heat Transfer: A Basic Approach*. McGraw-Hill, Inc., New York.

Pallas III, J.E. and Hamdy, M.K. 1976. Effects of thermoradiation on bacteria. Appl. Environ. Microbiol. 32(2):250-256.

Perkins, W.E., Ashton, D.H. and Evancho, G.M. 1975. Influence of the z value of *Clostridium botulinum* on the accuracy of process calculations. J. Food Sci. 40:1189-1192.

Perry, R.H. and Green, D.W. 1984. *Perry's Chemical Engineers' Handbook*, 6th ed. McGraw-Hill, Inc., New York.

Pohlman, A.J., Wood, O.B. and Mason, A.C. 1994. Influence of audiovisuals and food samples on consumer acceptance of food irradiation. Food Technol. 48(12):46-49.

Press, W.H., Teukolsky, S.A., Vetterling, W.T. and Flannery, B.P. 1992. *Numerical Recipes in FORTRAN*. Cambridge University Press, New York, NY.

Pszczola, D. 1993. Irradiated poultry makes U.S. debut in midwest and Florida markets. Food Technol. 47(11):89-96.

Reed, J.M., Bohrer, C.W. and Cameron, E.J. 1951. Spore destruction rate studies on organisms of significance in the processing of canned foods. Food Res. 16:383-408.

Sadeghi, F. and Swartzel, K.R. 1990. Generating kinetic data for use in design and evaluation of high temperature food processing systems. J. Food Sci. 55(3):851-853.

Sapru, V., Teixeira, A.A., Smerage, G.H. and Lindsay, J.A. 1992. Predicting thermophilic spore population dynamics for UHT sterilization processes. J. Food Sci. 57(5):1248-1252, 1257.

Schaffner, D.F., Hamdy, M.K., Toledo, R.T. and Tift, M.L. 1989. *Salmonella* inactivation in liquid whole egg by thermoradiation. J. Food Sci. 54(4):902-905.

Sedlak, M., Vinter, V., Ademec, J., Vohradsky, J., Voburka, Z. and Chaloupka, J. 1993. Heat shock applied early in sporulation affects heat resistance of *Bacillus megaterium* spores. J. Bacteriol. 175(24):8049-8052.

Shamsuzzaman, K. 1988. Effects of combined heat and radiation on the survival of *Clostridium sporogenes*. Radiat. Phys. Chem. 31(1-3):187-193.

Shamsuzzaman, K. and Lucht, L. 1993. Resistance of *Clostridium sporogenes* spores to radiation and heat in various nonaqueous suspension media. J. Food Protect. 56(1):10-12.

Shamsuzzaman, K., Payne, B., Cole, L. and Borsa, J. 1990. Radiation-induced heat-sensitivity and its persistence in *Clostridium sporogenes* spores in various media. Can. Inst. Food Sci. Technol. J. 23(2-3):114-120.

Shin, S. and Bhowmik, S.R. 1990. Computer simulation to evaluate thermal processing of food in cylindrical plastic cans. J. Food Eng. 12, 117-131.

Silva, C., Hendrickx, M., Oliveira, F. and Tobback, P. 1992. Optimal sterilization temperatures for conduction heating foods considering finite surface heat transfer coefficients. J. Food Sci. 57(3), 743-748.

Sivinski, H.D., Garst, D.M., Reynolds, M.C., Trauth Jr., C.A., Trujillo, R.E. and Whitfield, W.J. 1972. The synergistic inactivation of biological systems by thermoradiation. Ch. 17 in *Industrial Sterilization*, G.B. Phillips and W.S. Miller (ed.), p. 305-335. Duke University Press, Durham, North Carolina.

Skala, J.H., McGown, E.L. and Waring, P.P. 1987. Wholesomeness of irradiated foods. J. Food Protect. 50(2):150-160.

Stegeman, H., Mossel, D.A.A. and Pilnik, W. 1976. Studies on the sensitizing mechanism of pre-irradiation to a subsequent heat treatment on bacterial spores. In *Spore Research Vol. II*, A.N. Barker, J. Wolf, D.J. Ellar, G.J. Dring and G.W. Gould (ed.), p. 565-587. Academic Press, New York.

Stoer, J. and Bulirsch, R. 1993. *Introduction to Numerical Analysis*, 2nd ed. Springer-Verlag New York, Inc., New York.

Swartzel, K.R. 1982. Arrhenius kinetics as applied to product constituent losses in ultra high temperature processing. J. Food Sci. 47(6):1886-1891.

Swartzel, K.R. 1984. A continuous flow procedure for reaction kinetic data generation. J. Food Sci. 49(3):803-806.

Swartzel, K.R. 1986. Equivalent point method for thermal evaluation of continuous-flow systems. J. Agric. Food Chem. 34(3):396-401.

Swartzel, K.R. and Jones, V.A. 1985. System design and calibration of a continuous flow apparatus for kinetic studies. J. Food Sci. 50(4):1203-1204 & 1207.

Teixeira, A.A. 1992. Thermal process calculations. Ch. 11 in *Handbook of Food Engineering*, D.R. Heldman and D.B. Lund (ed.). p. 563-619. Marcel Dekker, Inc., New York.

Teixeira, A.A., Dixon, J.R., Zahradnik, J.W. and Zinsmeister, G.E. 1969. Computer optimization of nutrient retention in the thermal processing of conduction-heated foods. Food Technol. 23, 845-850.

Thayer, D.W. 1994. Wholesomeness of irradiated foods. Food Technol. 48(5):132-136.

Thayer, D.W., Songprasertchai, S. and Boyd, G. 1991. Effects of heat and ionizing radiation on *Salmonella typhimurium* in mechanically deboned chicken meat. J. Food Protect. 54(9):718-724.

Thibault, J. 1985. Comparison of nine three-dimensional numerical methods for solution of the heat diffusion equation. Num. Heat Trans. 8, 281-298.

Townsend, C.T., Esty, J.R. and Baselt, F.C. 1938. Heat resistance studies on spores of putrefactive anaerobes in relation to determination of safe processes for canned foods. Food Res. 3:323.

Walas, S.M. 1988. *Chemical Process Equipment*, Butterworth Publishers, Stoneham, MA.

Watts, D.G. 1994. Estimating parameters in nonlinear rate equations. Can. J. Chem. Eng. 72:701-710.

Welt, B.A., Steet, J.A., Tong, C.H., Rossen, and Lund, D.B. 1993. Utilization of microwaves in the study of reaction kinetics in liquid and semi-solid media. Biotechnol. Prog. 9:481-487.

Welt, B.A., Teixeira, A.A., Balaban, M.O., Smerage, G.H. and Sage, D.S. 1996. Iterative method for kinetic parameter estimation from dynamic thermal treatments. Unpublished manuscript. Agricultural and Biological Engineering Department, University of Florida, Gainesville, FL.

Welt, B.A., Teixeira, A.A., Lindsay, J.A. and Balaban, M.O. 1995. Kinetic parameter estimation from transient thermal treatments in conduction heating foods: Non-linear optimization. Presented at Ann. Mtg., Inst. of Food Technologists, Anaheim, Calif., June 3-7.

Welt, B.A., Tong, C.H., Rossen, J.L. and Lund, D.B. 1994. Effect of microwave radiation on inactivation of *Clostridium sporogenes* (PA 3676) spores. Appl. Environ. Microbiol. 60(2):482-488.

Welt, M.A. and Barnett, C.W. 1995. Automated irradiator for the processing of products and a method of operation. U.S. Patent 5,400,382.

Wescott, G.G., Fairchild, T.M. and Foegeding, P.M. 1995. *Bacillus cereus* and *Bacillus stearothermophilus* spore inactivation in batch and continuous flow systems. J. Food Sci. 60(3):446-450.

Whiting, R.C. and Buchanan, R.L. 1994. Microbial modeling. Food Technol. 48(6):113-120.

# BIOGRAPHICAL SKETCH

Bruce A. Welt was born April 21, 1967, in Raleigh, North Carolina. He received a Bachelor of Science degree in chemical engineering in 1989 at Clarkson University, Potsdam, New York. After graduation, He was employed by Alpha Omega Technology, Inc., a start-up food technology firm founded by his father, Martin A. Welt, Ph.D., now located in Cedar Knolls, New Jersey. After a little more than a year, he enrolled in the Food Science Department at Rutgers University, The State University of New Jersey, in New Brunswick, New Jersey. He received a Master of Science degree in 1993 from Rutgers University. Upon graduation, he was awarded a United States Department of Agriculture National Needs Fellowship to pursue a doctoral degree in food engineering at the University of Florida, Gainesville, Florida. He enrolled in the food engineering program with the Agricultural and Biological Engineering Department at the University of Florida in August, in 1993.

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Arthur A. Teixeira

Arthur A. Teixeira, Chairman
Professor of Agricultural and Biological
Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Murat O. Balaban
Associate Professor of Food Science and
Human Nutrition

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Glen H Smerage

Glen H. Smerage
Associate Professor of Agricultural and
Biological Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

David E. Hintenlang
Associate Professor of Nuclear
Engineering Sciences

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Burrell J. Smittle

Burrell J. Smittle
Professor of Entomology and
Nematology

This dissertation was submitted to the Graduate Faculty of the College of Engineering and to the Graduate School and was accepted as partial fulfillment of the requirements for the degree of Doctor of Philosophy.

August, 1996

fn Winfred M. Phillips
Dean, College of Engineering

Karen A. Holbrook
Dean, Graduate School